MCDONNELL DOUGLAS
INFORMATION SYSTEMS

Das Relationale Büro

ALL-Anwendungs-generator

Das Relationale Büro

ALL-Anwendungsgenerator

ALLGEMEINE BESCHREIBUNG

I N H A L T S V E R Z E I C H N I S

Kapitel/ Abschnitt	Oberschrift	Seite
1.0	A*L*L - WAS IST DAS ?	1.1
1.1	FÄHIGKEITEN DES SYSTEMS A*L*L	1.4
1.2	EINRICHTUNGEN UND VORTEILE DES SYSTEMS A*L*L	1.5
2.0	WIE ARBEITET DAS A*L*L-SYSTEM ?	2.1
2.1	SYSTEMRESSOURCEN-BEDARF	2.2
2.1.1	Hauptspeicherbelegung	2.2
2.1.2	Magnetplattenspeicher	2.3
2.1.3	Benutzerterminal	2.3
2.2	GEGENÜBERSTELLUNG: HERKÖMMLICHE PROGRAMMIERUNG - A*L*L	2.4
2.2.1	Vergleich: A*L*L - Produktivitätswerkzeuge	2.5
2.2.2	Vergleich der Leistungsfähigkeit	2.6
3.0	WIE WERDEN ANWENDUNGSSYSTEME MIT A*L*L ENTWICKELT ? ..	3.1
3.1	A*L*L-ARBEITSBLÄTTER	3.4
3.2	IM A*L*L-SYSTEM VERWENDETE TERMINOLOGIE	3.5
3.3	A*L*L GENERATORMENÜ	3.6
3.4	DATEIDEFINITION UND DATEN-DICTIONARY	3.7
3.4.1	Dateidefinition	3.7
3.4.2	Das globale Dictionary-System	3.8
3.4.3	Das Datei-Dictionary	3.9
3.5	DER FUNKTIONSGENERATOR	3.10
3.6	EINRICHTEN EINER BILDSCHIRMMASKE	3.10
3.6.1	1) Beispiel: Wartung der Kundendatei	3.11
3.6.2	2) Schrittweiser Aufbau der Bildschirmmaske	3.12
3.7	EIGENSCHAFTEN DES BERICHTGENERATORS	3.14
3.7.1	Aufbau eines Berichts mit A*L*L	3.15
3.7.2	Die für den Aufbau eines Berichts notwendigen Schritte	3.16
3.8	PFLEGE UND ÄNDERUNG DER BENUTZERFUNKTION	3.18
3.9	DOKUMENTATION MIT A*L*L	3.19
3.10	DIENSTROUTINEN DES A*L*L-SYSTEMS	3.20
3.11	DATENSCHUTZ-MENÜ	3.21

I N H A L T S V E R Z E I C H N I S

Kapitel/ Abschnitt	Oberschrift	Seite
4.0	SCHNITTSTELLEN ZWISCHEN A*L*L UND ANDEREN MICRODATA-SYSTEMEN	4.1
5.0	WEM NUTZT A*L*L ?	5.1
5.1.1	Anbieter und Systemhäuser	5.1
5.1.2	Datenverarbeitungs-Mitarbeiter	5.2
5.1.3	Fachbereichs-Benutzer	5.3
6.0	PRODUKTPOLITIK UND FIRMENGRUNDSÄTZE	6.1
6.1	MICRODATA-SOFTWARE	6.2
6.1.1	A*L*L-Software	6.2
6.1.2	Microdata System-Software	6.2
6.1.3	Optional erhältliche Software für die Serie M9000	6.2
6.1.4	System-Software der Serie M8000	6.4
6.1.5	Optional erhältiche Software-Systeme für die Serie M8000	6.4
7.0	A*L*L-SYSTEM - ZUSAMMENFASSUNG	7.1

1.0 A*L*L - WAS IST DAS ?

A*L*L ist eine Abkürzung für "Application Language Liberator" (frei übersetzt: "Befreier von Anwendungs-Programmmiersprachen"). Allein der Name verrät, daß es sich hierbei um ein revolutionäres und innovatives Konzept für die Entwicklung und Ausführung von Anwendungssystemen handelt. Mit A*L*L hat Microdata einen Weg beschritten, der absichtlich weg vom Einsatz herkömmlicher Programmiersprachen und Multisystem-Prozessoren führt. A*L*L befreit die Programmierer, Systemanalytiker und die technischen Mitarbeiter des Rechenzentrum von den zeitaufwendigen Routineaufgaben, der herkömmlichen Kodierung und dem Kompilieren von Programmen. A*L*L befreit die Benutzer außerdem von den üblichen Datenbank-Verwaltungsaufgaben. Durch A*L*L erhält der Anwendungssystem-Entwickler viel freie Zeit, in der neue Anwendungsprogramme entwickelt werden können, indem die normalerweise für die Erledigung von Routine-Verwaltungsaktivitäten benötigte Zeit wesentlich reduziert wird. Darüber hinaus ist automatisch gewährleistet, daß die gesamte Dokumentation stets auf dem neuesten Stand ist.

A*L*L ist eines der modernsten und fortschrittlichsten Werkzeuge für die Entwicklung von Anwendungssystemen, welches heute auf dem Markt erhältlich ist. Nach der Meinung der meisten Benutzer dieses Systems verdient A*L*L als inzwischen bewährtes System das Prädikat "Entwicklungs- und Ausführungssystem der 4. Generation".

A*L*L ist ein interaktives Entwicklungs- und Ausführungssystem, mit dem kommerzielle Anwendungssysteme implementiert, gewartet, dokumentiert und ausgeführt werden und welches besonders für das Definieren und Pflegen von Datenbank-Strukturen geeignet ist.

A*L*L

APPLICATION LANGUAGE LIBERATOR

UMFASST:

- IMPLEMENTIERUNG

- WARTUNG

- DOKUMENTATION

- AUSFÜHRUNG

VON KOMMERZIELLEN ANWENDUNGSSYSTEMEN

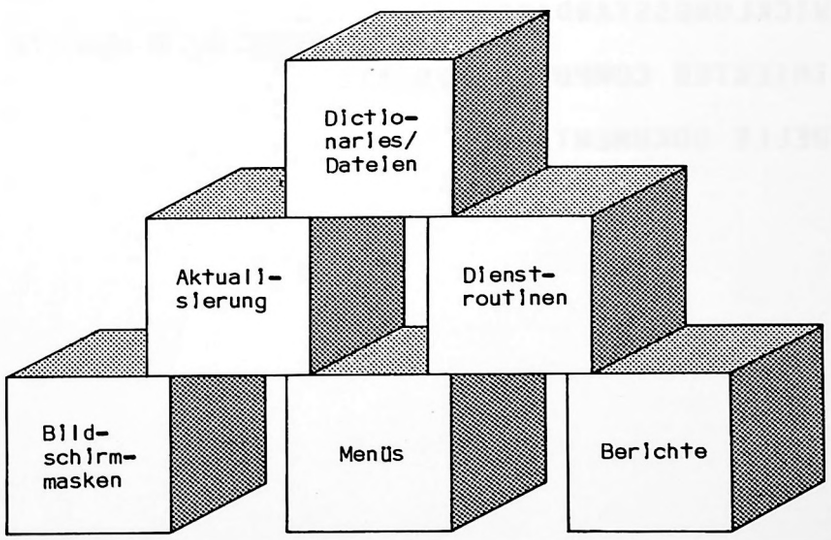
Mit dem System A*L*L erhält der Benutzer die Möglichkeit, integrierte Anwendungssysteme aufzubauen, Dateien zu definieren, ein übergeordnetes Dictionary zu erstellen und Bildschirmmenüs, Berichte und Aktualisierungsprogramme zu entwickeln.

Die dem System A*L*L unterliegende Technologie orientiert sich an ergonomischen, ökonomischen und benutzerfreundlichen Datenverarbeitungs-Methoden - ganz im Gegensatz zu vielen herkömmlichen Programmiersprachen. Es handelt sich hierbei um eine nicht prozedurale Vorgehensweise bei der Anwendungssystem-Entwicklung. Als Schnittstelle für den Anwendungssystem-Entwickler dient eine Reihe von Menüs und Bildschirmmasken, mit denen die Spezifikationen und Anforderungen der Benutzer definiert werden. Der Entwickler wird während der Entwicklungsphase mit Eingabeaufforderungen logisch durch die Systemparameter-Eingaben geleitet.

BIETET

INTEGRIERTE KOMMERZIELLE ANWENDUNGSSYSTEM-ENTWICKLUNG

ERSTELLT IHRE ANWENDUNGEN



.... OHNE HERKÖMMLICHEN PROGRAMMIERAUFWAND

A*L*L versteht einen Anwendungsprozeß als "Funktion". Der Begriff "Anwendungsprogramme" wird in Zusammenhang mit diesem System nicht benutzt, da A*L*L keine Programme erstellt. Vom Gesichtspunkt der Software-Entwicklung aus betrachtet, stellt A*L*L einen großen Schritt in die Richtung der automatisierten Anwendungssystem-Entwicklung dar.

Bei der Entwicklung des A*L*L-Systems wurde das Schwergewicht auf das Reduzieren der benötigten Betriebsmittel gelegt - Reduzieren der Kosten und Zeiten bei der Entwicklung von kommerziellen Anwendungssystemen, bei deren Modifikation, Dokumentation und Wartung.

A*L*L ist ein außergewöhnlich leistungsfähiges und kosteneffizientes Software-Produktivitätswerkzeug. Es ist ein Mittel, mit dem Anwendungssysteme und Datenbankstrukturen entsprechend den individuellen Bedürfnissen der Benutzer erstellt und modifiziert werden können.

A*L*L

- **ERHÖHTE PRODUKTIVITÄT**
  - **ENTWICKLUNG**
  - **MODIFIKATION**
  - **WARTUNG**
- **REDUZIERTE KOSTEN**
- **STRUKTURIERTE VORGEHENSWEISE**
- **ENTWICKLUNGSSTANDARDS**
- **OPTIMIERTER COMPUTER-EINSATZ**
- **AKTUELLE DOKUMENTATION**

1.1 FÄHIGKEITEN DES SYSTEMS A*L*L

Das System A*L*L bietet eine umfassende Lösung für die Entwicklung und Ausführung integrierter Anwendungssysteme. Der Entwickler ist in der Lage, benutzerindividuelle Datenbanken und Anwendungsmoduln aufzubauen. Die folgenden Hauptfunktionen werden im anschließenden Teil detaillierter beschrieben.

A*L*L

BIETET:

- **DATEI-DEFINITIONSMASKEN**
- **ÜBERGEORDNETE DATEN-DICTIONARIES**
- **INTERAKTIVE EINGABE-BILDSCHIRMMASKEN**
- **MENÜGESTEUERTE ANWENDUNGSSYSTEME**
- **KOMPLEXE BERICHTERSTATTUNG**
- **AKTUALISIERUNG BEI DER STAPELVERARBEITUNG**
- **HILFE- UND FEHLER-MELDUNGEN**
- **ARITHMETIK- UND LOGIK-OPERATIONEN**
- **DOKUMENTATION DER DATEIEN UND ANWENDUNGSSYSTEME**
- **MEHRERE DATENSCHUTZ-EBENEN**
- **SYSTEM-DIENSTPROGRAMME**

1.2 EINRICHTUNGEN UND VORTEILE DES SYSTEMS A*L*L

A*L*L verfügt über eine sehr große Anzahl von Einrichtungen; aus diesem Grund werden an dieser Stelle nur die wichtigsten aufgeführt.

- Die Wartung wird durch die standardisierte, strukturierte Vorgehensweise vereinfacht.

- Eine übergeordnete Datenbank mit Variablen-Namen kommt zum Einsatz, um die Konsistenz für alle Anwendungssysteme zu gewährleisten.

- Anwendungsfunktionen können während der Ausführung modifiziert werden.

- Die Anwendungssystem-Dokumentation für Dateien und Funktionen ist stets auf dem neuesten Stand.

- Es stehen Anwendungs-Dienstprogramme zur Verfügung, mit denen die Systementwicklung beschleunigt und die Systembibliothek gewartet wird.

- Menüs, interaktive Bildschirmmasken, Berichte und Aktualisierungsaktivitäten sind vollständig integrierte Teile des Entwicklungssystems. Der Einsatz nur eines Entwicklungssystems unterstützt die Standardisierung und beschleunigt die Entwicklung.

- Es ist möglich, die interaktive Verarbeitung von mehreren Anwendungssystemen ausgehend von mehreren Terminals auszuführen.

- Das System kann im Mandantenverfahren implementiert und eingesetzt werden; d.h., daß Daten firmen- oder abteilungsbezogen verarbeitet werden.

- Im Online-Verfahren können mehrere Dateien zusammen mit interaktiven Bildschirmmasken, Berichten oder Datenpflegeoperationen verarbeitet werden.

- Der Datenschutz-Mechanismus arbeitet auf mehreren Ebenen. Die Zugriffsberechtigungen können nach Anwendungssystem, Mitarbeiter, Terminal oder einer Kombination dieser Elemente definiert werden.

- Es ist möglich, standardisierte und benutzerindividuelle Prüfungen auf Datenfeld- und Datensatzebene auszuführen.

- Es steht ein umfangreicher Satz arithmetischer Operationen und bedingungsabhängiger Abfragen zur Verfügung.

- Die vom System und individuell vom Benutzer definierten Hilfe-Meldungen sind dem Benutzer während der Ausführung von Fehlerprüfungs- und Korrekturaktivitäten behilflich.

- Der modulare Funktionsentwurf erlaubt die Definition mehrerer logischer Funktionen mit eigenen Merkmalen innerhalb einer physischen Funktion.

- Die Dateneingabe wird durch verschiedene Optionen weitgehend optimiert. So ist es zum Beispiel möglich, Daten des unmittelbar zuvor verarbeiteten Datensatzes zu übernehmen, Datenfelder mit einem Standardwert zu vorzubesetzen und Tabulatorstops und Standard-Steuerzeichen zu setzen.

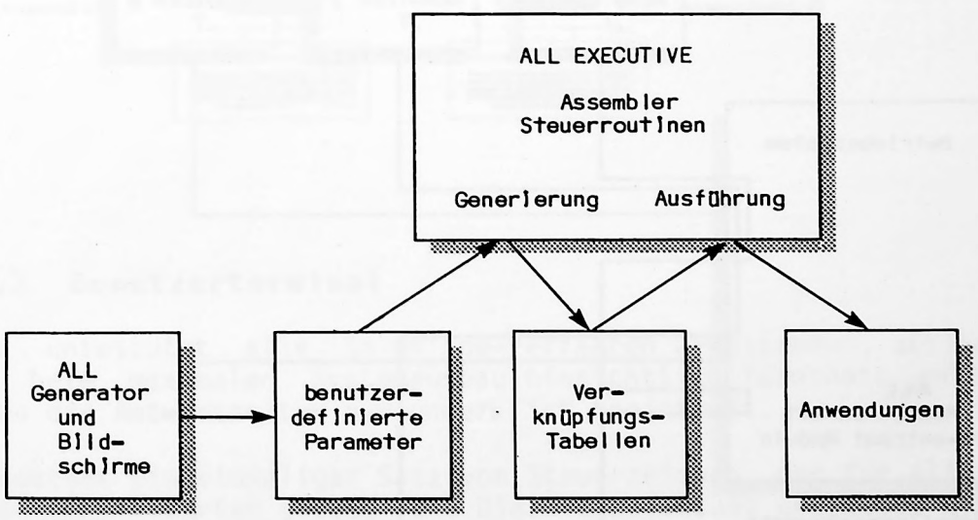
2.0 WIE ARBEITET DAS A*L*L-SYSTEM ?

Das dem A*L*L-System unterliegende Grundkonzept arbeitet darauf hin, daß für das Entwickeln eines kommerziellen Anwendungssystem bestimmte Datenverarbeitungs-Funktionen ausgeführt werden müssen. A*L*L ist ein Nukleus sehr leistungsfähiger, strukturierter und äußerst effizienter Makro-Routinen, die in Assembler geschrieben sind. Diese Makro-Routinen umfassen das gesamte Spektrum der Datenverarbeitungs-Aufgaben oder funktionsorientierter Prozesse. A*L*L ist in dem Betriebssystem REALITY vollständig integriert und verwendet die von diesem Betriebssystem gebotenen Dienstleistungen.

Der A*L*L-Anwendungsgenerator ist in ein Entwicklungssystem und ein Ausführungssystem aufgeteilt. Das Entwicklungssystem kommt während der Anwendungsentwicklung zum Einsatz, wenn der Benutzer die benötigten Parameter definiert. Das Generieren der Funktionen ist mit dem Kompilierprozeß fast identisch, mit dem Unterschied, daß keine Programme aufgebaut werden und nur sehr wenig Zeit (gemessen in Sekunden) in Anspruch genommen wird. Das Entwicklungssystem baut aus den vom Benutzer zur Verfügung gestellten Parametern eine Verknüpfungstabelle auf. Diese Tabelle ist der logische Pfad durch die verschiedenen Makro-Instruktionen oder Ausführungsroutinen. Bei den Ausführungsroutinen des Systems A*L*L handelt es sich um tabellengesteuerte Routinen, bei denen nur die für die Ausführung der jeweiligen Funktion benötigten Moduln angesprochen werden. Die optimerten Assembler -Routinen sind re-entrant; d.h. sie werden nur einmal im Hauptspeicher abgelegt, können jedoch von mehreren Anwendern gleichzeitig gebraucht werden.

APPLICATION LANGUAGE LIBERATOR

KONZEPTANALYSE



2.1 SYSTEMRESSOURCEN-BEDARF

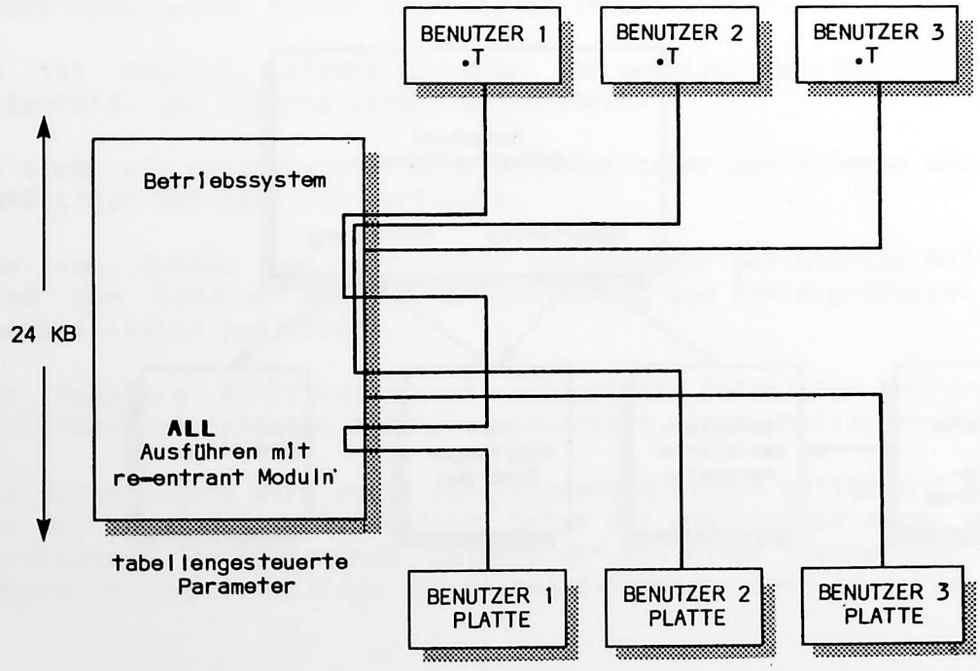
2.1.1 Hauptspeicherbelegung

Der gesamte A*L*L-Anwendungsgenerator (d.h. Entwicklungs- und Anwendungssystem) ist in einem Hauptspeicherbereich von maximal 24 KB untergebracht. Dieser Overhead kann jedoch gedanklich vernachlässigt werden, da zu einem gegebenen Zeitpunkt nur ein Teil der Routinen für die Entwicklung und Ausführung benutzt wird, was selbstverständlich abhängig von den Bedürfnissen des jeweiligen Anwendungssystems ist. Das gesamte System unterliegt einer dynamischen Ressourcen-Zuordnung.

Unabhängig davon, ob es sich bei dem Benutzer um einen Anwendungssystem-Entwickler oder um einen Mitarbeiter der Datenerfassung oder eines Fachbereichs handelt, ist stets nur eine Kopie des jeweils benötigten, wiederverwendbaren (re-entrant) Moduls im Hauptspeicher vorhanden, der von allen Systembenutzern gemeinsam eingesetzt wird. Statt für jeden Benutzer jeweils ein Programm in den Hauptspeicher zu laden oder jedem Entwickler einen großen Prozessor exklusiv zu reservieren, ist nur ein Programm für alle Benutzer vorhanden, wodurch wertvolle Hauptspeicherkapazität eingespart wird.

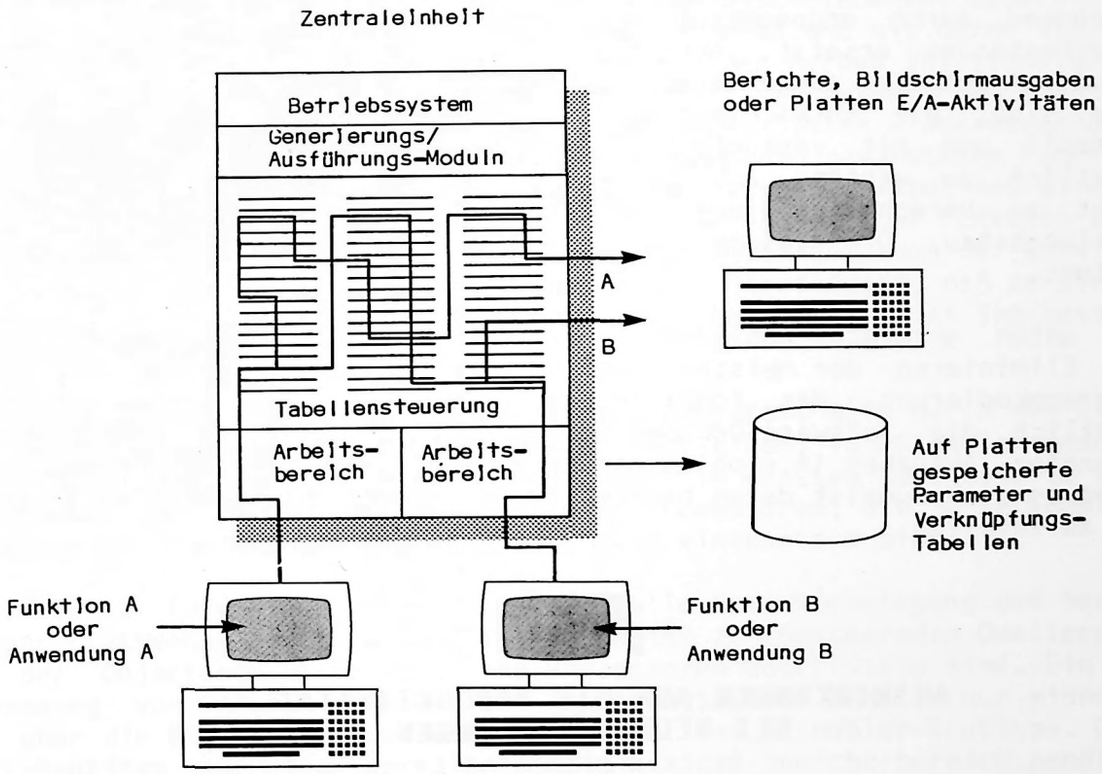
Während der Ausführung ist die Verknüpfungstabelle im Vergleich zu großen, herkömmlichen Objektprogrammen sehr klein. Die Tabelle ist bis zu 1 KB groß. Durch diese optimale Hauptspeicherbelegung ist mehr Platz im Hauptspeicher für die eigentliche Datenverarbeitung vorhanden.

AUSFÜHRUNG WIEDERVERWENDBARER MODULN



2.1.2 Magnetplattenspeicher

Wie auch das Betriebssystem REALITY unterstützt A*L*L variabel lange Felder, Datensätze und Dateien, wodurch die zur Verfügung stehende Plattenspeicher-Kapazität optimal genutzt wird. Diese Verarbeitungsmethode steht im krassen Gegensatz zur Verarbeitung mit festen Feld- und Datensatzlängen, bei der wesentlich mehr Plattenspeicherbereiche in Anspruch genommen wird. Anstatt große Quellprogramme zu speichern, bewahrt A*L*L nur die vom Benutzer eingegebenen Parameter auf, mit denen das Anwendungssystem und die Verknüpfungstabellen definiert werden. Die Verknüpfungstabelle ist im Vergleich zu großen, herkömmlichen Quellprogrammen sehr klein. A*L*L benötigt nur sehr wenig Plattenspeicherbereiche, was zur Folge hat, daß größere Benutzer-Datenblöcke im Hauptspeicher vorhanden sein können.



2.1.3 Benutzerterminal

A*L*L untestützt alle im Online-Verfahren arbeitenden, aktiven Terminals. Auch beim maximalen Systemausbau hinsichtlich Terminals und Hauptspeicher werden die Antwortzeiten nur unmerklich beeinflußt.

Es besteht ein einmaliger Satz von Steuerzeichen, der für alle unterschiedlichen Terminalarten gültig ist. Diese Einrichtung gestattet die problemlose und mühelose Kommunikation zwischen den Mitarbeitern der Datenerfassung und der Fachbereiche und dem Anwendungssystem-Entwickler.

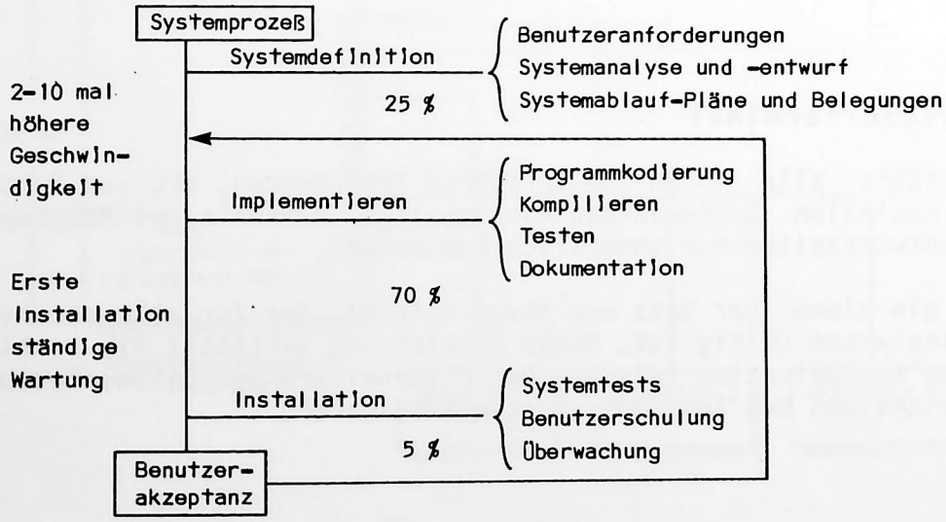
2.2 GEGENOBERSTELLUNG: HERKÖMMLICHE PROGRAMMIERUNG - A*L*L

Bei genauerer Betrachtung des A*L*L-Systems wird der ungeheuere Einfluß auf die Produktivitätskrise deutlich. Mit dem Einsatz dieses Systems ist der Anwendungssystem-Entwickler in der Lage, Anwendungssysteme viel schneller aufzubauen, als dies unter Verwendung herkömmlicher Programmiersprachen oder Programmgeneratoren möglich wäre. Die folgende Tabelle zeigt, daß Anwendungssysteme mit A*L*L zwei bis zehn mal schneller als mit anderen Systemwerkzeugen erstellt werden können. Diese Abbildung befaßt sich nur mit der Neuentwicklung; wenn zusätzliche Wartungsaufgaben eingebettet werden, lassen sich noch größere Unterschiede erkennen. Selbstverständlich hängen die prozentualen Einsparungen im starken Maße von den jeweiligen Arten von Anwendungssystemen ab.

Anwendungsgeneratoren der vierten Generation, wie z.B. A*L*L, stellen die Software-Lösungen der Zukunft dar. Die konventionelle Programmierung wird zunehmend durch ergonomisch gestaltete Software-Systeme bei wachsenden Datenbeständen ersetzt. Aktuelle Statistiken belegen, daß derzeitig nicht genug, am Computer ausgebildete Mitarbeiter zur Verfügung stehen, die in der Lage sind, die Entwicklung der auf dem Markt erhältlichen Computersysteme sinnvoll und mit vernünftigen zeitlichen Vorstellungen voranzutreiben und letztlich zu warten. Aus diesem Grund spricht man in der Computer-Branche nicht zu Unrecht von einem sehr ernst zu nehmenden Software-Anwendungsentwicklungsstau. Die Lösung zu diesen Problemen liegt im Einsatz des A*L*L-Systems.

Die unten gezeigte Abbildung verdeutlicht die Zeiteinsparungen, die durch das Eliminieren der meisten überflüssigen Aufgaben entstehen, wie z.B. die Programmkodierung, das Kompilieren, Testen und die Fehlerkorrektur und letztlich die Software-Dokumentation. Das A*L*L-System kombiniert alle genannten Aufgaben in problemlos und schnell auszuführenden Prozeduren, bei denen es sich zumeist darum handelt, leere Felder mit sinnvollen Informationen aufzufüllen.

AUSWIRKUNGEN AUF DIE PRODUKTIVITÄT BEI NEUENTWICKLUNGEN



2.2.1 Vergleich: A*L*L - Produktivitätswerkzeuge

Die für die Steigerung der Produktivität angebotene Palette reicht von Programmgeneratoren über parameter-gesteuerte Produkte, Berichterstattungs-Generatoren, Bildschirmmasken-Generatoren bis hin zu Dienstroutinen, die alle für die Erledigung bestimmter Aufgaben entwickelt wurden. Viele dieser Werkzeuge steigern die Programmierereffektevität erheblich. Zumeist sind diese Werkzeuge jedoch nur begrenzt einsetzbar und benötigen zusätzliche Programmkodierung. In einigen Fällen lassen sie sich jedoch überhaupt nicht einsetzen, nämlich dann, wenn die individuellen Benutzerbedürfnisse zu stark vom üblichen Rahmen abweichen. Programmgeneratoren erzeugen eine große Menge kodierter Instruktionen, die letztlich kompiliert werden müssen; ein Umstand, der häufig zu nicht effizient ablaufenden Anwendungssystemen führt.

Das A*L*L-System ist der Kern von im Assembler geschriebenen, äußerst effizienten Maschinencode-Routinen, die reentrant sind und mit denen die Funktionen zur Verfügung gestellt werden, die für den Aufbau vollständig integrierter Anwendungssysteme notwendig sind. Statt der Auswahl von Makros und der Entwicklung von Programmen benutzt das A*L*L-System wiederverwendbare, über Parameter gesteuerte Routinen, mit denen die gewünschten Funktionen oder Aufgaben ausgeführt werden. Der Ablauf der Assembler-Routinen wird über die Verknüpfungstabellen gesteuert. Bei einer Änderung einer Funktion ändert der Entwickler lediglich die Parameter der Tabellen. Wenn der Entwickler erst einmal mit A*L*L vertraut ist, ist es überflüssig, daß er sich mit einer Vielzahl unterschiedlicher Prozessoren befaßt, um eine ihm gestellte Aufgabe zu lösen.

Die Leistungsfähigkeit des gesamten Computersystems wird durch den Einsatz des A*L*L-Systems optimiert, da A*L*L nur mit einem, in Assembler geschriebenem Primärprogramm arbeitet. Dies steht im krassen Gegensatz zu vielen Programmen und häufig unterschiedlichen Prozessoren, die im herkömmlichen Verfahren für die Entwicklung und Ausführung einzusetzen sind.

A*L*L sorgt für eine effiziente, virtuelle Speicherbelegung und benötigt weniger Plattenkapazität, da die üblicherweise zu speichernden Quellprogramme, der Objektcode und zusätzliche Prozessoren überflüssig sind. Ein unter Verwendung von A*L*L entwickeltes Anwendungssystem benutzt nur einen Teil der über die Bibliothek zur Verfügung gestellten Assembler-Routinen. Da die A*L*L-Routinen nur einen verhältnismäßig kleinen Speicherbereich benötigen, wird die Speicherbereichbelegung des virtuellen und echten Speichers weitgehend optimiert, was auch zu Verbesserungen der Durchsatzraten und der Gesamtsystemleistung führt.

2.2.2 Vergleich der Leistungsfähigkeit

A*L*L bietet eine moderne, innovative Gesamtsystem-Lösung, die im Wettbewerb mit Programmgeneratoren, über Parameter gesteuerte Software-Produkte, Berichterstattungs-Generatoren und Bildschirmmasken-Generatoren steht.

Es wurden viele Vergleichstest ausgeführt, mit denen es möglich ist, einen Überblick über die Leistungsfähigkeit des A*L*L-Systems zu gewinnen. Dabei stellte sich heraus, daß A*L*L den Vergleich mit den verschiedensten, herkömmlichen Programmierverfahren und angebotenen Produktivitätswerkzeugen nicht zu scheuen braucht.

So z.B. wurde in Verbindung mit einem Kreditorensystem ein Vergleichstest ausgeführt, bei dem die Dienstprogramme des Betriebssystems und eine herkömmliche Programmiersprache zum Einsatz kamen. Bei herkömmlicher Vorgehensweise wurden 100 Stunden mehr als bei A*L*L benutzt. In dem Vergleichstest konnte eine Effektivitätssteigerung von 85 % festgestellt werden.

Erst durch A*L*L wird die benutzerindividuelle Anwendungsentwicklung realistisch. Die Amortisationsphase des Systems ist sehr kurz; das heißt, daß die Schnittstelle zwischen den entstehenden Kosten und den bei der Anwendungsentwicklung und -wartung eingesparten Kosten sehr schnell erreicht ist.

A*L*L

ENTWICKLUNGS-VERGLEICHSTEST

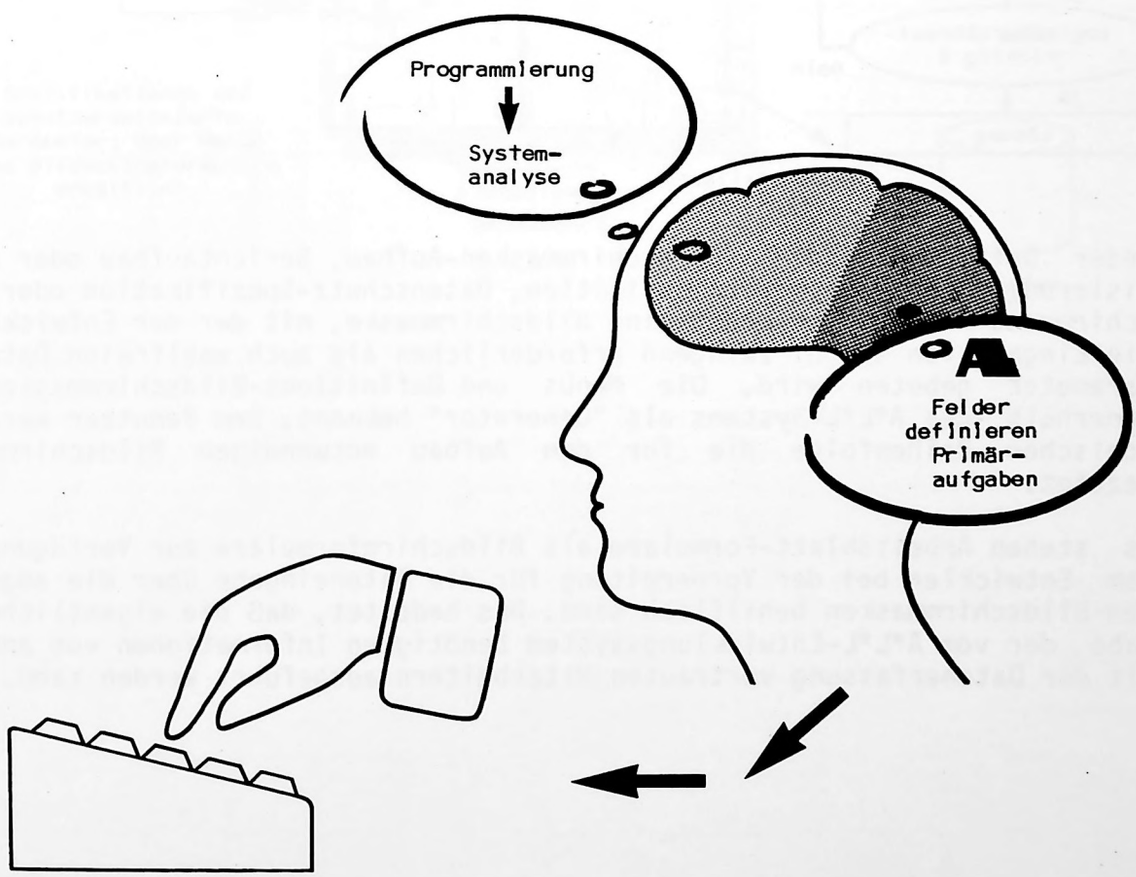
ANWENDUNGSSYSTEM	MANNSTUNDEN BEI HERKÖMMLICHER SYSTEMENTWICKLUNG	MANNSTUNDEN BEI SYSTEMENTWICKLUNG MIT A L L	PRODUKTIVITÄTS-STEIGERUNG MANNSTD.	%
Kreditorenmenü	1,50	0,25	1,25	83
Lieferantenwartung	24,00	1,00	23,00	96
Lieferschein-Eingabe	80,00	16,00	64,00	80
Sequentielle Lieferschein-Auflistung mit Update	8,00	0,50	7,50	94
Lieferanten-Auflistung	5,00	0,25	4,75	95
Lieferanten-Auflistung mit SPEED!	0,25	0,25	-,--	--
	118,25	18,25	100,50	85%

3.0 WIE WERDEN ANWENDUNGSSYSTEME MIT A*L*L ENTWICKELT ?

Bei dem Entwurf und der Entwicklung des A*L*L-Systems wurde besonders auf ergonomische und benutzerfreundliche Technologien geachtet, wobei von der Voraussetzung ausgegangen wurde, daß die bisher üblichen Programmiersprachen diese Kriterien nicht erfüllen. Der Entwurf und die Entwicklung mit A*L*L bedeutet, daß von oben nach unten gearbeitet wird; eine Vorgehensweise, bei der der Entwickler zuerst den allgemeinen Rahmen definiert und erst dann Detailinformationen zur Verfügung stellt.

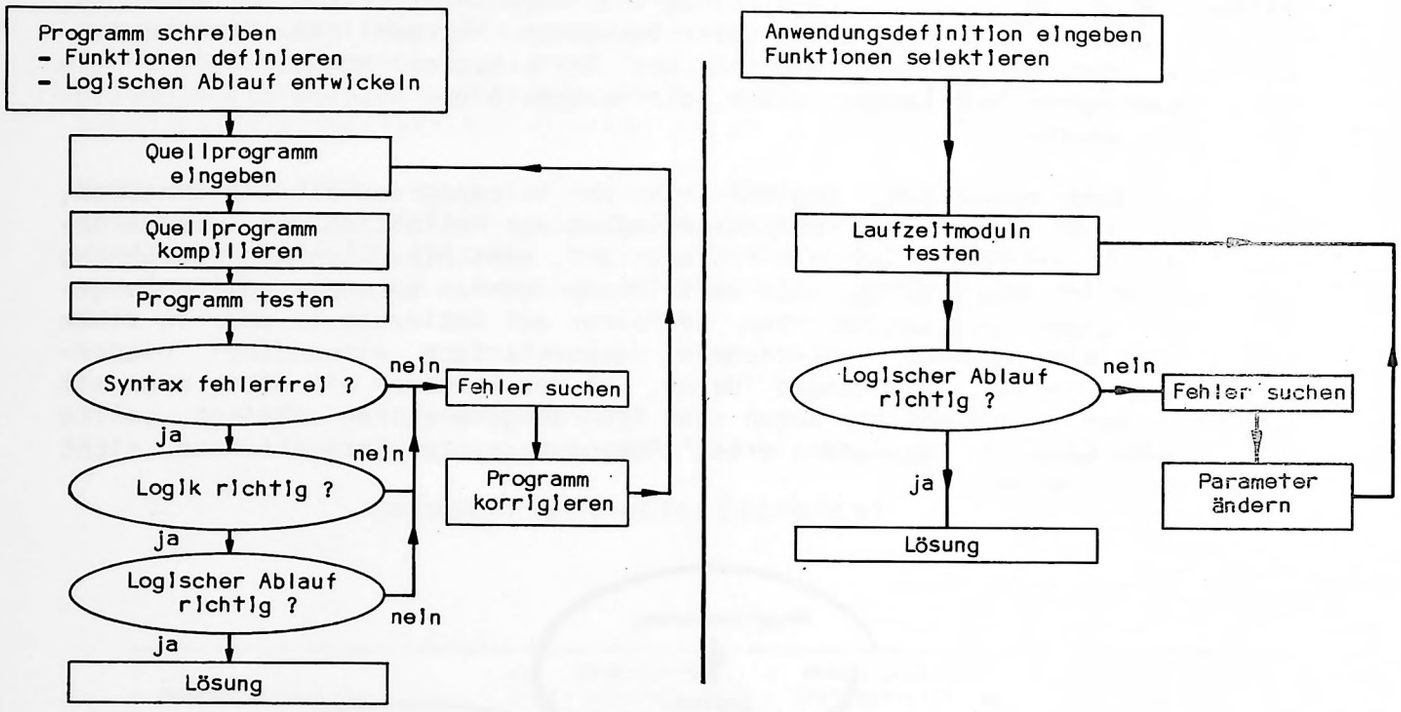
Der Anwendungsentwurf ist in dieser Vorgehensweise die erste Phase. Dies trifft nicht nur auf die Entwicklung von Anwendungssystemen mit A*L*L zu, sondern auch auf die Entwicklung unter Verwendung herkömmlicher Programmiersprachen oder Programmgeneratoren. Das A*L*L-System standardisiert die Definition der Anwendungsaufgaben, die innerhalb des Systems als "Funktionen" bekannt sind.

Die Anwendungsentwicklung beginnt mit der Belegung von Bildschirmmasken, Berichten und Dateien und geht anschließend zur Definition der Sekundärdateien und zur Datenerfassungs-Prüfung über, einschließlich der Ausführung arithmetischer Operationen. Die Definitionen können entweder völlig losgelöst von einem standardisierten Verfahren auf Notizzetteln oder in einem mehr ordnungsliebendem Unternehmen dokumentarisch einwandfrei niedergeschrieben werden. Unabhängig davon, ob der Benutzer mit A*L*L oder mit herkömmlichen Programmiersprachen oder Programmgeneratoren arbeitet, sollte die (vorzugsweise standardisierte) Anwendungssystem-Entwurfsphase nicht übersprungen werden.



A*L*L geht erst in der Anwendungs-Entwicklungsphase einen anderen Weg, besonders bei dem Aufbau der Datenbankstruktur und dem Implementieren und Testen des Anwendungssystems. Anstatt eine häufig mit Fehlern behaftete Kodierung des anstehenden Problems durchzuführen, werden dem Entwickler einige Menüs zur Verfügung gestellt, deren Leerfelder entsprechend den individuellen Spezifikationen der Benutzer auszufüllen sind.

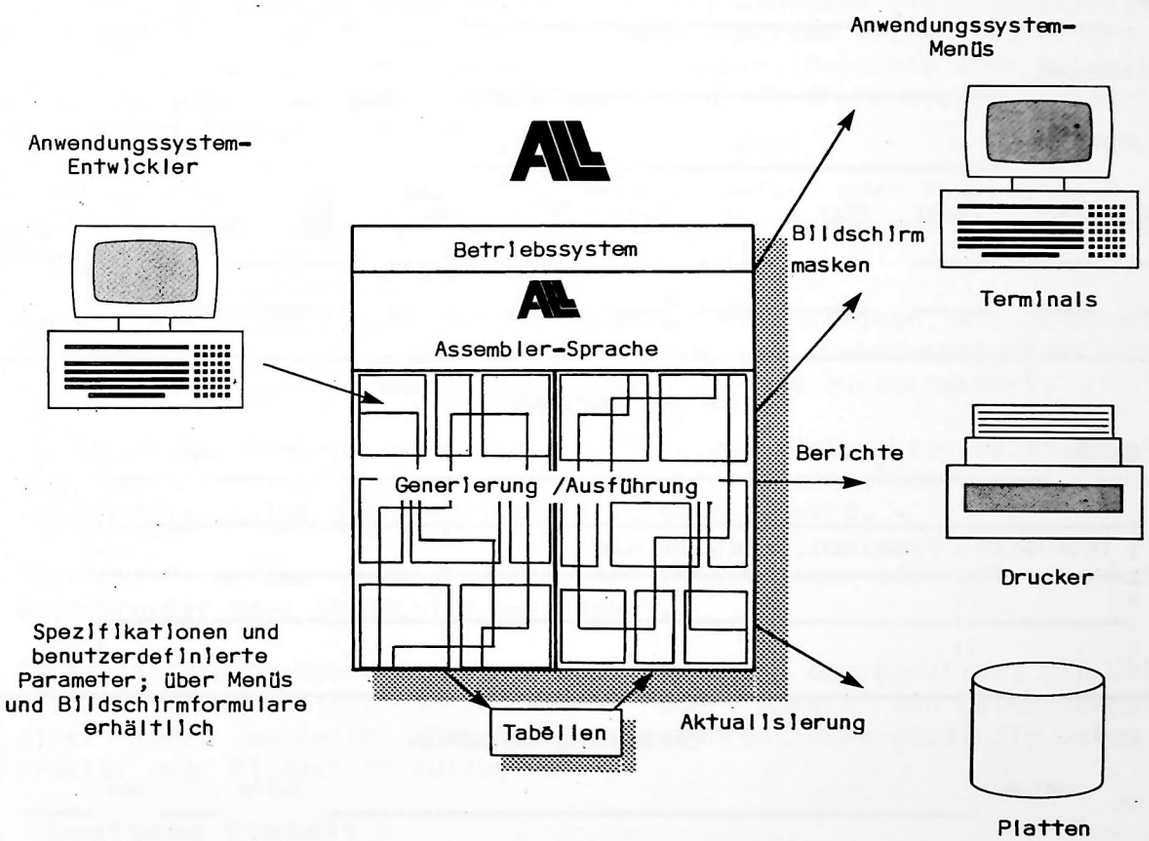
HERKÖMMLICHE PROGRAMMIERUNG **... EINSATZ VON A*L*L**



Jeder Definitionsprozeß (Bildschirmmasken-Aufbau, Berichtaufbau oder Aktualisierungs-Operation, Dateidefinition, Datenschutz-Spezifikation oder Bildschirmmenü-Aufbau) verwendet eine Bildschirmmaske, mit der der Entwickler um die Eingabe von sowohl zwingend erforderlichen als auch wahlfreien Daten und Parameter gebeten wird. Die Menüs und Definitions-Bildschirmmasken sind innerhalb des A*L*L-Systems als "Generator" bekannt. Dem Benutzer werden in logischer Reihenfolge die für den Aufbau notwendigen Bildschirmmasken gezeigt.

Es stehen Arbeitsblatt-Formulare als Bildschirmformulare zur Verfügung, die dem Entwickler bei der Vorbereitung für die Dateneingabe über die angezeigten Bildschirmmasken behilflich sind. Das bedeutet, daß die eigentliche Eingabe der vom A*L*L-Entwicklungssystem benötigten Informationen von anderen, mit der Datenerfassung vertrauten Mitarbeitern ausgeführt werden kann.

Der Anwendungssystem-Entwickler gibt die Spezifikationen und die vom Benutzer definierten Parameter über mehrere, menügesteuerte Eingabe-Bildschirmmasken ein, um die benötigten Funktionen zu definieren. Nach der Parametereingabe werden diese Informationen einem Generierungsprozeß zugeführt. Der Generierungsprozeß ist für den Aufbau der Parametertabelle verantwortlich. Die entstandene Tabelle wird während der Ausführung der Funktionen oder Anwendungssystem-Aufgaben eingesetzt. Die in der Tabelle zu findenden Einträge stellen den logischen Ablauf durch die auszuführenden Routinen dar. Der Anwendungsentwickler kann als Funktionstyp ein Menü, eine Bildschirmmaske, einen Bericht oder eine Aktualisierung spezifizieren.



3.1 A*L*L-ARBEITSBLÄTTER

Der Anwendungssystem-Entwickler kann die unten gezeigten Arbeitsblätter bei der Definition der vom Benutzer gewünschten Funktionen und bei der Spezifikation der individuellen Anwendungssystem-Kriterien einsetzen. In den meisten Fällen wird der Entwickler nur die Formulare "Menü- und Bildschirmformate" und "Bildschirmfelder" benutzen. Muster der Arbeitsblätter sind unten gezeigt.

MENÜ- UND BILDSCHIRMFORMATE

DATUM: _____ Seite _____ von _____

FUNKTIONSNAME: _____

FORMATID: _____

FOLGE	ZEILE	SPALTE	TEXT	FOLGE	ZEILE	SPALTE	TEXT
_____	_____	_____	_____	_____	_____	_____	_____
_____	_____	_____	_____	_____	_____	_____	_____

TERMINALZUORDNUNGEN

DATUM: _____ Seite _____ von _____

TERMINALID	FIRMA/ABTL.	BESCHREIBUNG

DATENSCHUTZ-KATEGORIEN

DATUM: _____ Seite _____ von _____

FIRMA/ABTL.	BENUTZERIDENTIFIK.	KATEGORIE	EBENE

B I L D S C H I R M F E L D E R

DATUM: _____ Seite _____ von _____

FUNKTIONSNAME: _____

FELD	NAME	ZEILE	SPL	LÄNG	FÜLL CODE	AUSG FORM	SOND PRÜF	M J	N C	D O	A R	O V	T B	LGC ID

3.2 IM A*L*L-SYSTEM VERWENDETE TERMINOLOGIE

Bei den meisten, im Zusammenhang mit dem A*L*L-System verwendeten Begriffe, handelt es sich um die üblicherweise verwendete Terminologie; es gibt jedoch einige Begriffe, die möglicherweise nur A*L*L-spezifisch interpretiert werden sollten. Die Basisterminologie (z.B. Feld, Datensatz und Datei) ist selbsterklärend. Selbstverständlich werden auch die Begriffe "Schlüsselfeld", "Datensatzschlüssel", "alphanumerisches Feld" und "numerisches Feld" benutzt.

1. Funktion

Eine Funktion ist mit einer Anwendungssystem-Aufgabe gleichzusetzen, kann jedoch auch als Anwendungssystem-Modul oder Programm betrachtet werden. Als Funktionen gelten Menüs, Eingabe-Bildschirmmasken, Berichte oder Aktualisierungs-Operationen. Der Begriff "Programm" wird mit A*L*L nicht verwendet, da keine Programme generiert werden.

Das A*L*L-System teilt Anwendungssystem-Aufgaben oder Funktionen in vier Kategorieren auf: Menüs, Bildschirmmasken, Berichte und Aktualisierungs-Operationen.

a) Ein "Menü" bietet eine Liste auswählbarer Funktionen, mit denen es dem Benutzer möglich ist, einen bestimmten Teil des Anwendungssystems für die Ausführung anzuwählen wonach das System zum Menü zurückzukehrt.

b) "Bildschirmmasken" erlauben es dem Benutzer, interaktiv Daten zu erfassen, Dateien zu pflegen und andere Funktionen auszuführen, bei denen der interaktive Dialog über Terminal vorausgesetzt wird.

c) "Berichte" werden dazu benutzt, den Inhalt einer oder mehrerer Dateien über Drucker oder Bildschirm auszugeben.

d) "Aktualisierungs-Operationen" ermöglichen es dem Benutzer, die übliche Aktualisierung (Hinzufügen, Löschen oder Ändern von Datensätzen) auf einer (oder mehreren) Datei(en) auszuführen, ohne dabei die Daten über Drucker oder Bildschirm auszugeben.

2. Logische Einheit

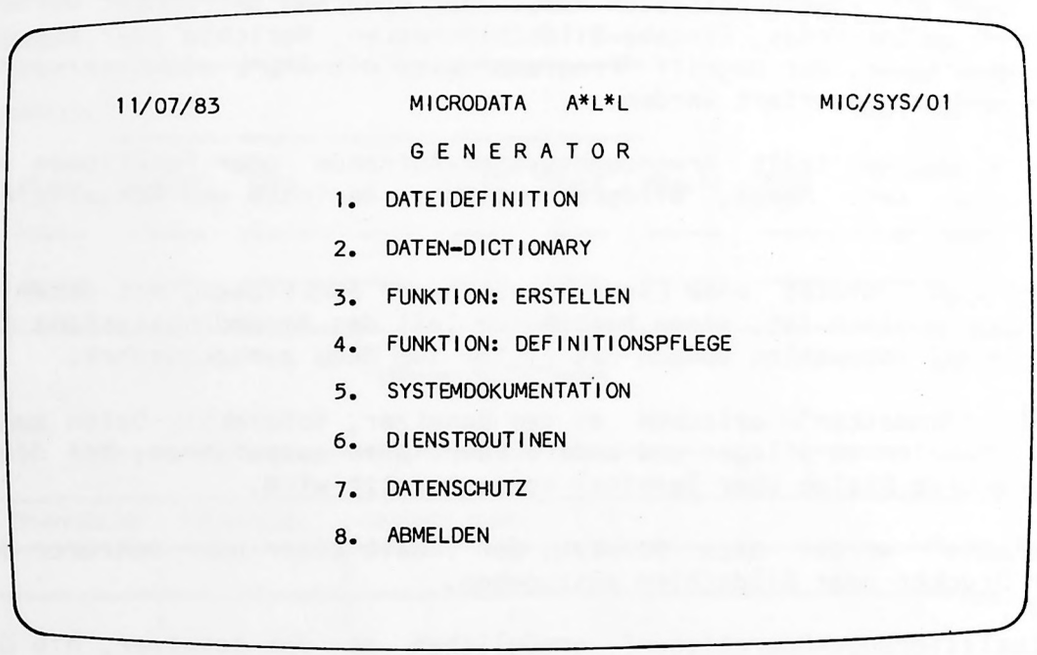
Das System betrachtet eine Bildschirmmaske, einen Bericht oder eine Aktualisierungsfunktion als physische Einheit, die sich aus logischen Einheiten oder "Fenstern" zusammensetzt. Diese logischen Einheiten arbeiten ein- oder mehrmalig (Blätterfunktion oder Paging).

Bei einmaliger Arbeitsweise besteht eine direkte Beziehung zwischen Datensatz und logischer Einheit.

Bei mehrmaliger Arbeitsweise werden mehrere Datensätze mit einer logischen Einheit verknüpft.

3.3 A*L*L GENERATORMENÜ

Das A*L*L Generatormenü dient als Startpunkt für das Erstellen und Modifizieren von Anwendungssystemen und für andere verfügbare Prozessoren. Das Generatormenü bittet den Benutzer um die Eingabe einer Ziffer, mit der der gewünschte Prozessor angesteuert werden muß. Das A*L*L Generatormenü ist unten dargestellt.



3.4 DATEIDEFINITION UND DATEN-DICTIONARY

Dieser Abschnitt befaßt sich mit den ersten zwei Auswahlmöglichkeiten des A*L*L Generatormenüs; "Dateidefinition" und "Daten-Dictionary"; der Teil des Systems, der für die Definition der Daten angesprochen werden muß.

3.4.1 Dateidefinition

Innerhalb des A*L*L-Systems werden verschiedene Funktionen oder Anwendungssystem-Aufgaben eingesetzt, mit denen Daten eingegeben, geändert und ausgegeben werden. Um zu erreichen, daß die Funktionen die Daten ordnungsgemäß verarbeiten, müssen die Daten nach bestimmten Formaten organisiert und definiert werden.

Mit "Dateidefinition" werden die Datei, deren Datensätze und die darin enthaltenen Felder oder Dateivariablen definiert. Dies ist der erste Schritt im Aufbau eines Anwendungssystems.

Nachdem die Dateidefinition abgeschlossen ist, aktualisiert A*L*L automatisch die Variablen-Bibliothek. Wenn nach dieser Definition ein Feld oder eine Variable ausgehend von einer Bildschirmmaske, einem Bericht oder einer Aktualisierungs-Operation aufgerufen wird, prüft das System den Namen und greift automatisch auf die in diesem Pool zu findenden Standardeinträge zu. Sind vom Endbenutzer bestimmte Wünsche zu berücksichtigen, bei denen die in diesem Pool gespeicherten Standardeinträge nicht verwendet werden sollen, kann der Entwickler den Inhalt dieser Felder überschreiben.

Bei der unten gezeigten Abbildung handelt es sich um eine Dateidefinitions-Bildschirmmaske. Es wird eine Kundendatei mit 12 Standardeinträgen im übergeordneten System-Dictionary erstellt.

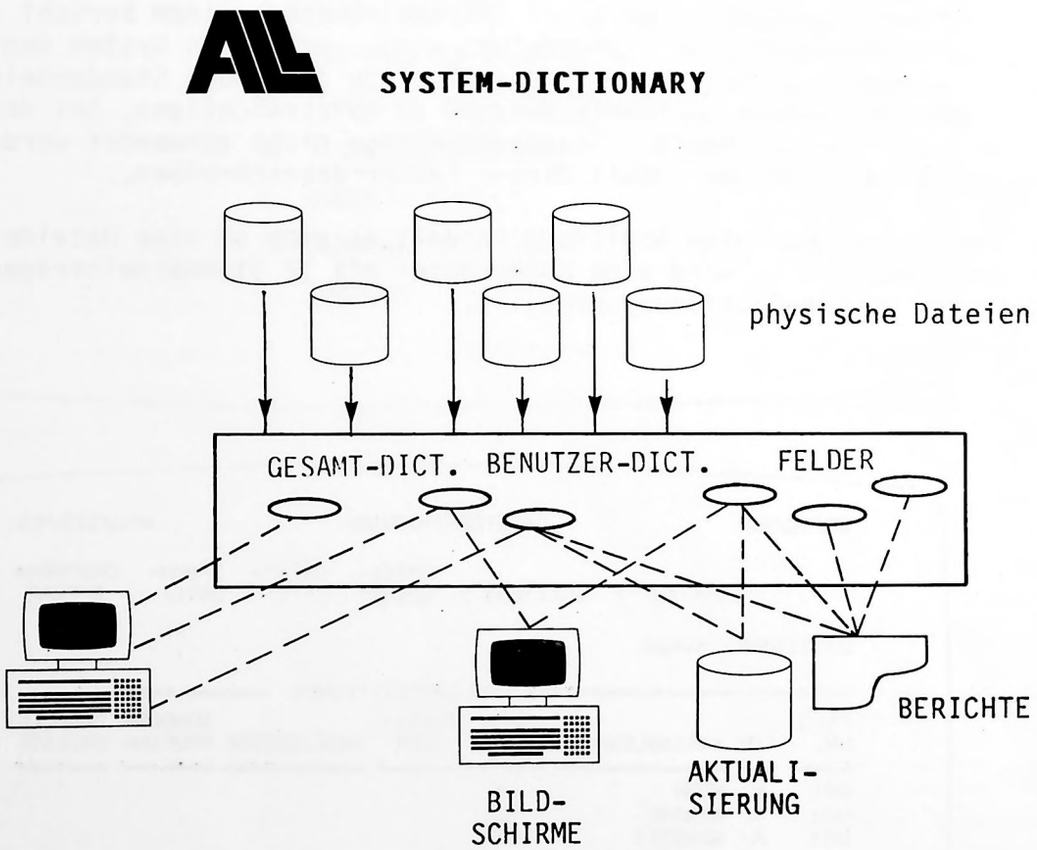
```
05/10/84                    DATEIDEFINITION                    MIC/SYS/03

                                          DATEI-   UNTER-   STAMM-   GRUPPEN-
        FIRMA/ABT ?  BLÄTTERN ?           SPERRE   DATEI    DATEI     SPERRE

DATEINAME: KUNDE

------------------------- FELDDEFINITIONEN -------------------
FELD                          MAX.  FUELL-                SONDER-  TABELLE
NR.   TYP  VARIABLENNAME     LAENGE  CODE   ANZEIGECODE   PRÜFUNG  GROESSE
----  ---  -------------     ------  ------  -----------  -------  -------
001    K   KDNR                 6
002    A   KDNAME              30
003    A   KDADDR1             30
004    A   KDADDR2             30
005    A   KDPLZORT            30
006    A   KDLKZ                5
008    A   KDTEL               10
009    A   KDTELDIR             5
010    N   KDKONTAKT           20
011    N   KDSALDO             10          2
012    N   KDRECHDAT            6          DATE         DATE
013    N   KDBETRAG            10          2
```

3.4.2 Das globale Dictionary-System

A*L*L ist ein über das System-Dictionary gesteuertes Produkt, welches über eine äußerst leistungsfähige Datenbank-Struktur verfügt und für die Standardisierung in allen Teilen der Entwicklungsphase eines Anwendungssystem-Moduls sorgt.

Die Standardeinträge enthalten das Datenformat und die maximale Länge eines Feldes. Der Füllcode gibt an, ob es sich hierbei um ein links- oder rechtsbündig, mit Nullen oder Leerzeichen vor oder nach den Daten aufzufüllendes Feld handelt. Das Ausgabeformat gibt an, wie das Feld bei der Ausgabe aufbereitet werden soll. Es gibt Standard- und Sonder-Anzeigecodes, die einem jeden Datenfeld zugeordnet werden können. Es können entweder Standardprüfungen oder benutzerindividuell kodierte Prüfungen ausgeführt werden. Darüber hinaus ist es möglich, eine bis zu 70 Zeichen große Hilfe-Meldung dem Standardeintrag hinzuzufügen, auf der man bei der Eingabe zugreifen kann.

Das folgende Diagramm erläutert das Konzept des "System-Dictionary" in schematischer Darstellung.



3.4.3 Das Datei-Dictionary

Innerhalb des A*L*L-Systems werden vier unterschiedliche Arten von Variablen verwendet. Bei der ersten Art handelt es sich um die "Globaldatei-Variable", die im Dateidefinitions-Prozeß verwendet wird. Die zweite Art ist die "Wertvariable", eine global gültige, funktionsunabhängige Art. Bei der dritten Variablenart handelt es sich um eine "Systemvariable", die über das Gesamtsystem Gültigkeit hat (z.B. aktuelles Systemdatum und Uhrzeit). Die vierte und letzte Variablenart ist die "temporäre Variable", die nur in Verbindung mit der jeweiligen Funktion Gültigkeit hat und nicht auf Platte gespeichert wird.

Es steht ein Dictionary-Verwaltungsmenü (ein Untermenü) zur Verfügung, welches mit den Systemteilen arbeitet, die für das Definieren und die Pflege der unterschiedlichen Arten der Variablen innerhalb des A*L*L-Systems verantwortlich sind. Die unterschiedlichen Variablenarten können unter Verwendung des unten gezeigten Untermenüs "Dictionary-Pflege" (Dictionary Maintenance) ausgedruckt werden.

05/10/84 M I C R O D A T A A * L * L MIC/SYS/03

DICTIONARY-PFLEGE

AUSWAHL NUMMER	KURZBEZEICHNUNG	FUNKTIONS-NAME
1.	SPEEDI-DICTIONARIES	@DICT
2.	SYSTEM-DICTIONARIES	@FVAR
3.	DATEI PHYSISCH ERSTELLEN	@CREATEF
4.	DATEI/FUNKTION QERVERBINDUNGSPRÜFUNG	@FXREF
5.	WERTVARIABLEN	@VAR
6.	WERTVARIABLEN-DOKUMENTATION	@VARDOC
7.	ANZEIGEN SYSTEMDATEI	@SYSLOOK
8.	FUNKTIONS-VERZEICHNIS	@FUNDIR

3.5 DER FUNKTIONSGENERATOR

Es ist möglich, unter Verwendung des A*L*L-Funktionsprozessors "ERSTELLEN" neue Funktionen zu erstellen. Dabei führt der Prozessor den Entwickler mittels Generator-Bildschirmmasken durch die einzelnen Stadien der Entwicklungsphase. Abhängig von der Art der gewünschten Funktion schafft der Prozessor die Verknüpfung zur benötigten Entwicklungs-Eingabemaske des A*L*L-Systems. Beim Einsatz dieses Teils des Systems wird die Entwicklungszeit erheblich verkürzt, bei gleichzeitiger Ausschaltung der ansonsten üblichen Fehlerquellen.

3.6 EINRICHTEN EINER BILDSCHIRMMASKE

- Es ist möglich, Bildschirmmasken interaktiv zu entwickeln und zu pflegen.

- Der Datenzugriffsschutz kann auf mehreren Ebenen definiert werden: Funktion, Kategorie, Benutzer und Terminal.

- Mit einer Bildschirmmaske können mehrere Dateien verarbeitet werden.

- Es können standardisierte und benutzerindividuelle Prüf- und Editierroutinen auf Feldern entweder feldspezifisch oder datensatzspezifisch ausgeführt werden.

- Der Entwickler ist in der Lage, umfangreiche arithmetische und bedingungsspezifische Operationen zu definieren, die sich entweder feld- oder datensatzspezifisch auf die Verarbeitung auswirken.

- Es können bis zu neun logische Hintergrundformate oder -bildschirmmasken für eine Funktion verwendet werden.

- Es wird der modulare Aufbau von Bildschirmmasken unterstützt, der es erlaubt, eine Vielfalt logischer Funktionen mit jeweils eigenen Merkmalen innerhalb einer physischen Funktion zu definieren.

- Durch die Verfügbarkeit von Einrichtungen, wie z.B. die automatische Übernahme von Daten aus dem vorangegangenen Datensatz, die Übernahme von Standardwerten und der Einsatz von Tabulatoren-Stops und Standard-Steuerzeichen, wird die Datenerfassung optimiert.

- Es stehen die im System vorhandenen und gesondert vom Benutzer definierten Hilfe-Meldungen für die operationelle Unterstützung und Benutzerführung bei Fehlerprüf- und Fehlerkorrektur-Aktivitäten zur Verfügung.

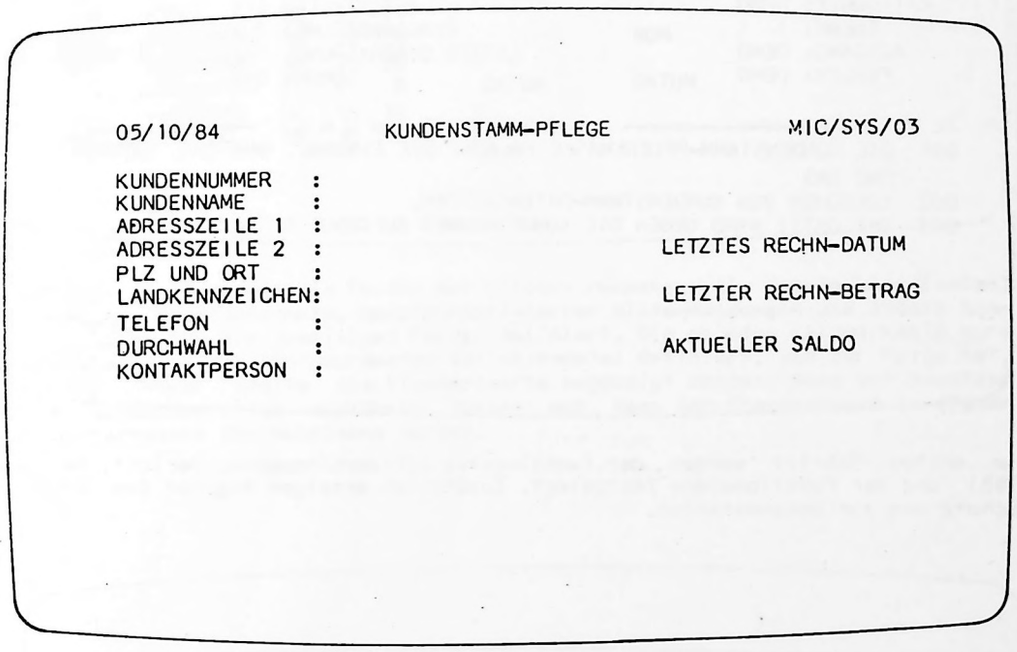
- Es können benutzerindividuelle und standardisierte Anzeigecodes für jedes einzelne Datenfeld definiert werden, so daß es möglich ist, jedes Feld entsprechend den Bedürfnissen der Benutzer zu formatieren.

- Jedem einzelnen Datenfeld können Füllcodes hinzugefügt werden, mit denen die Werte von Datenfeldern automatisch links- oder rechtsbündig mit Leerzeichen oder Nullen aufgefüllt werden.

- Standardprüfungen bezüglich der Datenfelder könnten wie folgt lauten: "Mußeingabe-Feld", "keine Änderung zum Feld" oder "nur Anzeigen".

3.6.1 1) Beispiel: Wartung der Kundendatei

Um zu zeigen, wie mühelos z.B. eine Kundenstammdatei-Bildschirmmaske aufgebaut und erstellt werden kann und wie einfach und strukturiert die Vorgehensweise ist, wenden wir uns nun der Entwicklung eines Teils eines Anwendungssystems zu. Ein Beispiel der Kundeninformations-Bildschirmmaske ist unten dargestellt.



3.6.2 2) Schrittweiser Aufbau der Bildschirmmaske

Für den Aufbau einer Bildschirmmaske zur Kundeninformation sind nur vier Definitionsbildschirme zu beachten. Die Definitionen werden in den nächsten vier Abbildungen gezeigt und kurz erklärt.

```
05/10/84                    FUNKTIONSDEFINITION                    MIC/SYS/03

FUNKTIONSNAME: KUNDPF      TITEL: KUNDENSTAMMDATEI-PFLEGE
         TYP: S
    KATEGORIE: DEMO                          ANALYTIKER: SYS
       EBENE: 1                        EINGABEDATUME: 20/02/83
     AUSGANG: DEMO                  LETZTE GENERIERUNG: 11/03/83 11:49:32
      FEHLER: DEMO                              DURCH: SYS

  ZL # ------------------  B E S C H R E I B U N G  ----------------
  001  DIE KUNDENSTAMM-PFLEGEMASKE ERLAUBT DIE EINGABE, WARTUNG, ABFRAGE
       UND DAS
  002  LOESCHEN VON KUNDENSTAMM-DATENSAETZEN.
  003  DIE DATEI WIRD UEBER DIE KUNDENNUMMER AUFGERUFEN.
```

Im ersten Schritt werden der Funktionstyp (Bildschirmmaske, Bericht, Menü, Typ) und der Funktionsname festgelegt. Zusätzlich erfolgen Angaben zum Datenschutz und zur Dokumentation.

```
05/10/84          MENUE-/BILDSCHIRM-FORMAT DEFINITION          MIC/SYS/03

FUNKTIONSNAME: KUNDPF              KUNDENSTAMMDATEI-PFLEGE
 FORMATNUMMER: 1

                          1.........2.........                  1..... .2......
FOL ZEI SPAL 0....5....0....5....0....5...9 SPAL 0..5..0...5. .0..5..9
GE  LE   TE  ---------- TEXT ---------- TE  ------- TEXT --------
 01   2 : 28  KUNDENSTAMM-PFLEGE
 02   4 :  4  KUNDENNUMMER          :          :
 03   6 :  4  KUNDENNAME            :          :
 04   8 :  4  ADRESSZEILE 1         :          :
 05  10 : 11  ZEILE 2               :          :
 06  12 :  4  PLZ UND ORT           :          :
 07  14 :  4  LANDKENNZEICHEN       :          :
 08  16 :  4  TELEFON               :          :
 09  18 :  4  DURCHWAHL             :          :
 10  20 :  4  KONTAKTPERSON         :          :
 11  10 : 55  LETZTES RECHN-DATUM   :          :
 12  13 : 55  LETZTER RECHN-BETRAG  :          :
 13  17 : 55  AKTUELLER SALDO       :          :
```

Schritt 2 gibt dem Entwickler vollkommen freie Hand bei der Eingabe und Änderung von Bildschirmformaten und -masken. Dieser Prozess erlaubt auch den Aufbau mehrerer Bildschirmmasken für einen Bildschirm.

```
05/10/84          BILDSCHIRMMASKEN-FELDER DEFINITION          MIC/SYS/03

FUNKTIONSNAME: KUNDPF          KUNDENSTAMMDATEI-PFLEGE

FELD  NAME       ZEI  SPAL LNG FÜLL ANZEIGECODE  SONDER GEN  M N D A O T LGC
NR.              LE   TE       CODE              PRÜF   PRÜF I C O R V R ID
----  ---------  ---  ---- --- ---- ----------  ------ ---- - - - - - - ---
001   KDNR         4   22   6  LFN                        J
002   KDNAME       6   22  30
003   KDADDR1      8   22  30
004   KDADDR2     10   22  30
005   KDPLZORT    12   22  30
006   KDLKZ       14   22   5
007   KDTEL       16   22  11
008   KDTELDIR    18   22   5                     NUM
009   KDKONTAKT   20   22  20
010   KDRECHDAT   11   61   6       DATUM        DATUM
011   KDBETRAG    14   56  10       2
012   KDSALDO     18   56  10       2                          J
```

Im Schritt 3 werden die Felder der Bildschirmmasken, die Anzeigekoordinaten, Prüfungen, Anzeigeformate, benutzerdefinierten Hilfe-Meldungen und andere Sonderoptionen für die jeweiligen Felder definiert. Die meisten Feldmerkmale wurden bereits in der übergeordneten Variablendatei definiert, was zur Folge hat, daß an dieser Stelle die Standardwerte angezeigt werden. Wenn der Benutzer eine Sonderoperation ausführen lassen muß, kann der Standardwert in dieser Bildschirmmaske überschrieben werden.

```
05/10/85          BILDSCHIRMATTRIBUTE DEFINITION          MIC/SYS/03

FUNKTIONSNAME: KUNDPF    KUNDENSTAMMDATEI-PFLEGE

---------- BILDSCHIRMATTRIBUTE-INFORMATIONEN ----------
SCHIRM # : 1      LS-MODUS              : ACDL
FORMATNR.: 1      MODUSZEILE            :          BLAETTERN          :
LOESCHEN : J      STARTFELD #           :  1       STARTZEILE #       :
EIN MAL  :        ENDFELD #             : 12       ENDZEILE #         :
LOGIKNR. :        LETZTES GELESENES FELD:  1       ANZ. ANZEIGEZEILEN:

---------- DATEIINFORMATIONEN ----------
                  DATEIMODI  DATEIZUGRIFF   LOGIKNR  NACH LESEN VOR
DATEI# -- NAME -- A C D L  RD-FLD RTN-FLD            FEHL/KEIN SCHRE
 01    KUNDE                  1      1
```

Mit Schritt 4 werden die Bilschirmmasken-Struktur, die zugeordneten Dateien und Modusoptionen der Bildschirmfunktionen eingegeben und geändert. Während der Bildschirmmasken-Verarbeitung können verschiedene Arten von Logikroutinen ausgeführt werden.

3.7 EIGENSCHAFTEN DES BERICHTGENERATORS

- Es ist einfach, benutzerindividuelle Berichte zu entwickeln und zu warten.

- Der Bericht kann auf mehreren Ebenen geschützt werden: Anwendung, Funktion, Mitarbeiter, Terminal und Unternehmen bzw. Abteilung.

- Die Berichtsüberschriften können konstante oder variable Daten enthalten, wobei die variablen Daten von normalen Dateien oder Systemdateien übernommen werden.

- Es ist möglich, Seitenüberschriften, logische Berichtüberschriften oder Überschriften beim Erkennen eines Gruppenwechsels zu definieren.

- Mit den A*L*L-Berichterstattungsmoduln lassen sich Sonderformulare verarbeiten, wie z.B. Rechnungsformulare, Lohn-/Gehaltsstreifen, usw..

- Es können benutzerindividuelle und standardisierte Anzeigeformate für jedes einzelne Datenfeld definiert werden, so daß es möglich ist, jedes Feld entsprechend den Bedürfnissen der Benutzer zu formatieren.

- Es ist möglich, auf vertikaler Ebene ein Summenkennzeichen zu setzen, welches dafür verantwortlich ist, automatisch die Summe eines bestimmten Datenfeldes zu erstellen.

- Umfassende arithmetische und bedingungsspezifische, logische Operationen können feld- oder datensatzspezifisch und vor oder nach der Verarbeitung eines Gruppenwechsels ausgeführt werden.

- Es können beliebige Datenfelder in auf- oder absteigender Reihenfolge sortiert werden.

- Es ist möglich, für einen jeden Bericht Sortier- und Auswahlkriterien anzugeben.

- Bei der Erstellung des Berichts können mehrere Dateien verarbeitet werden.

- Der Teil einer Variablen kann als Kriterium für einen Gruppenwechsel dienen.

3.7.1 Aufbau eines Berichts mit A*L*L

In den folgenden Seiten befassen wir uns mit einem Beispiel, um zu demonstrieren, wie mühelos ein Bericht erstellt werden kann - in diesem Fall die Kundenstammdaten. Es sollte besonders beachtet werden, daß bestimmte Ähnlichkeiten zwischen den Bildschirmformularen für die Definition der Bildschirmmasken und den Bildschirmformularen zur Definition des gewünschten Berichtes zu verzeichnen sind. Ob nun ein Menü, eine Bildschirmmaske, ein Bericht oder ein Aktualisierungsprozeß entwickelt werden soll - die Kontinuität des Aufbaus und der Struktur ist im gesamten System gewährleistet. Die folgende Abbildung zeigt ein Beispiel des Kundenstammdatei-Berichts.

14:27:25 05/10/84 MICRODATA CORPORATION SEITE 1

K U N D E N S T A M M D A T E I - L I S T E

KUNDEN NR.	KUNDENNAME TELFON	ADRESSZEILEN	PLZ STADT UND LAND	LETZTE RECHNUNG	AKTUELLER SALDO
000001	VONDOR GMBH 08503-23890	POSTFACH 112 HAUPTSTR. 11	D-6945 HIRSCHBERG	06/07/83 250.75	375.90
000002	SAUER-VERTRIEBS GMBH 089-2768901	BAYERNSTR. 77	D-8000 MUENCHEN 11	12/08/83 566.00	985.89
000003	CONTUR AG 0211-78452	POSTFACH 4000 NECKARSTR. 73	D-2000 DUESSELDORF 1	23/09/83 1000.70	1540.45
000004	SUESSLI 072-678	KANTONSTR. 12	CH-8048 ZUERICH	07/10/83 856.80	895.70

FORDERUNGS-BESTAND: 3797.94

KUNDEN-GESAMTUMSATZ 26.215,87

3.7.2 Die für den Aufbau eines Berichts notwendigen Schritte

Für das Erstellen eines Kundenstammdatei-Berichts sind fünf Schritte auszuführen. Dafür werden die folgenden Bildschirmformulare benutzt.

```
05/10/84                    FUNKTIONSDEFINITION                    MIC/SYS/03

FUNKTIONSNAME: KUNDLI    TITEL: KUNDENSTAMMDATEI-AUFLISTUNG
         TYP: R
    KATEGORIE: DEMO                        ANALYTIKER: SYS
        EBENE:                          EINGABEDATUM: 05/10/84
    AUSSPRUNG: DEMO                GRUNDSTELLUNG AM: 05/10/84  15:14:56
       FEHLER: DEMO                             DURCH: SYS

ZL # ---------------  B E S C H R E I B U N G  ----------------------
001  KUNDENSTAMMDATEI-AUFLISTUNG
```

Im ersten Schritt werden der Funktionstyp (Bildschirmmaske, Bericht, Menü, Typ) und der Funktionsname festgelegt. Zusätzlich erfolgen Angaben zum Datenschutz und zur Dokumentation.

```
05/10/84              BERICHTKOPF  DEFINITION                MIC/SYS/03

FUNKTIONSNAME: KUNDLI                    KUNDENSTAMMDATEI-AUFLISTUNG
   KOPFIDENT.: P
                        1.........2.........
FOL ZEI A SPAL 0....5....0....5....0....5...9
GE  LE  B  TE  ----------  TEXT  ----------  VAR?  LNG  ANZEIGECODE
 01  1 :    1  @ZEIT                          J    - 8  ZEIT
       :   12  @DATUM                         J      8  DATUM
       :   49  $FIRMNAME                      J     30
       :   88  SEITE
       :   94  @SEITE                         J      3  0
 02  2 :   27  K U N D E N
       :   42  S T A M M
       :   55  D A T E I  -
       :   70  L I S T E
 03  2 :    1  KUNDEN
       :    9  KUNDENNAME
       :   76  LETZTE
       :   87  AKTUELLER
 04  1 :    3  NR.
       :    9  TELEFON
       :   35  ADRESSZEILEN
       :   52  PLZ STADT UND LAND
       :   76  RECHNUNG
       :   87  SALDO
 05  1 :    1  ←                             J     66
       :   67  ←                             J     30
```

```
05/10/84              BERICHTKOPF  DEFINITION              MIC/SYS/03

FUNKTIONSNAME: KUNDLI                    KUNDENSTAMMDATEI-AUFLISTUNG
   KOPFIDENT.: 1

                                1.........2.........
FOL ZEI A SPAL 0....5....0....5....0....5....0....5...9
GE  LE  B  TE  ----------  TEXT  ----------  VAR?  LNG  ANZEIGECODE
 01  1 :   65  FORDERUNGS-BESTAND
```

Schritt 2 gestattet die Eingabe und Änderung aller Berichtkopfzeilen. Die Kopfzeilen enthalten sowohl Konstanten als auch variable Informationen, die aus normalen Dateien und Systemdateien entnommen werden. Die Kopfzeilen können am Anfang einer jeden Seite, nur am Berichtanfang oder am Anfang nach dem Gruppenwechsel gedruckt werden.

```
05/10/84              BERICHTFELDER-DEFINITION              MIC/SYS/03

FUNKTIONSNAME: KUNDPF              KUNDENSTAMMDATEI-PFLEGE

-------------------------------------------------------------------------
FELD  ----NAME----  ZEI  A  SPAL  LNG  ANZEIGECODE  LOGIK-  VERT
NR.                 LE   B  TE                      ID      SUMM
----  ------------  ---  -  ----  ---  ----------  ------  ----
001   KDNR           1       1     6
002   KDNAME                 9    23
003   KDADDR1                35   14
004   ŁLKZ.PLZ.ORT           52   22
005   KDRECHDAT              76    8   DATUM
006   KDSALDO                84   10   2                        J
007   KDTEL          1       10   14
008   KDADDR2                35   14
009   KDBETRAG               73   10   2
```

Im Schritt 3 werden die im Bericht auszugebenden Felder sowie die Druckkoordinaten, Druckformate, Anzeigecodes, vertikal errechnete Summen und andere Sonderattribute der Felder definiert.

```
05/10/84                    BERICHT-DEFINITION                    MIC/SYS/03

FUNKTIONSNAME: KUNDLI      KUNDENSTAMMDATEI-AUFLISTUNG

------------------  LOGISCHE BERICHTINFORMATIONEN  -----------------
BERICHT #      : 1     STARTFELD #:  1   HÖHE    : 56 SORT/AUSWAHL:  N
KOPFID         :       ENDFELD #  :  9   BREITE  : 99 ABSTEIGEND  :
EIN MAL ?      : N     BLÄTTERN ? :  N   Z/SATZ  :  4 S/A LGC-ID  :
LOGIK-ID       :                                      S/A EDATEI #:
SORTIEREN NACH: KUNDSALDO

-------------------------  DATEIINFORMATIONEN  ---------------------
                 MODUS CLR WECHSEL LOGIKNR  VOR  NACH LESEN VOR
DATEI#-- NAME --                            LESEN FEHL/KEIN SCHRE
 01    KUNDE        L                                   1


------------------  GRUPPENSTEUERUNG  DEFINITION  ------------------
CB-VARIABLE-VAR-POS DRK LOGIK-NR ZEILENSPRUNG KOPF START
NR---NAME--- ST END SUM VOR NACH  VOR   NACH   ID  SPALTE BEZ.KONSTANTE
1    &ŁEOF            J            1           1
```

Der Schritt 4 erlaubt es dem Entwickler, die Struktur und Arbeitsweise des Berichtgenerators einzugeben oder zu ändern. Jeder logische Bericht kann damit definiert werden, unabhängig davon, ob "geblättert" werden soll, ob auf- oder absteigend zu sortieren ist oder ob Auswahlkriterien und Gruppenwechsel-Operationen zum Einsatz kommen sollen.

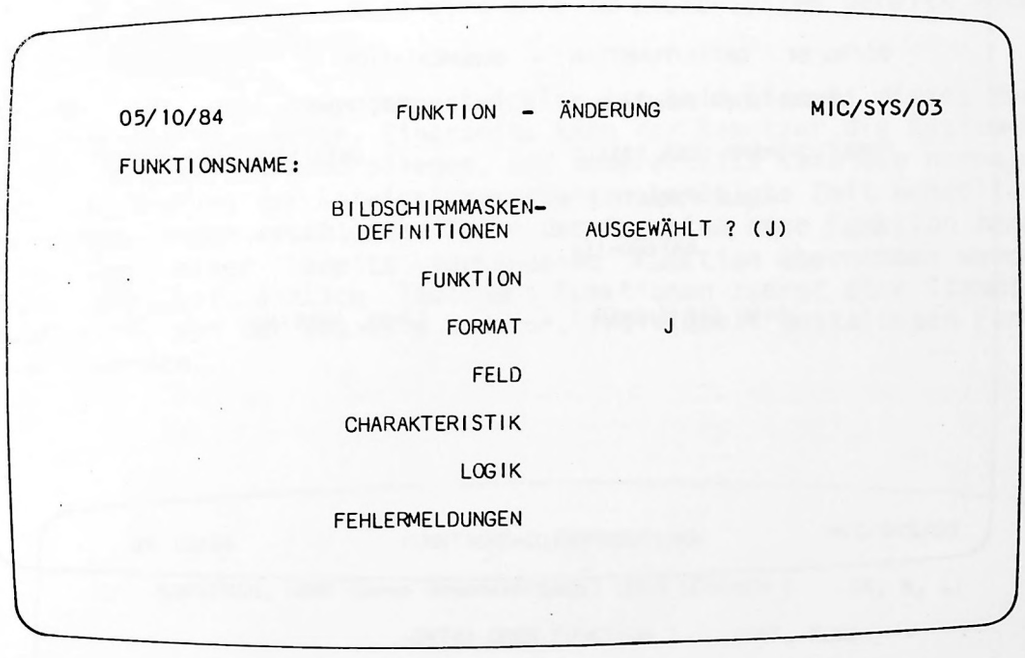
```
05/10/84                          LOGIKDEFINITION                MIC/SYS/03

FUNKTIONSNAME: KUNDLI              TITEL: KUNDENSTAMMDATEI-AUFLISTUNG
    LOGIK-NR.: 001

ZEILE   SCHRITT   --------------      ANWEISUNG      --------------
 001       *      BERICHT: 1   DATEI: 1
 002       *      VERKETTUNG VON PLZ, ORT, LKZ
 004              ŁLKZ.PLZ.ORT = KDLKZ + "-" + KDPLZORT
```

Der Schritt 5 ist für die Eingabe und Pflege von benutzerspezifischen Berechnungen und Bedingungsroutinen bestimmt. Der Logikteil wird für nicht standardisierte Prüfungen, Kalkulationen und für den Aufbau von Auswahlkriterien für die Verarbeitung benötigt. Der Logikteil ist wahlfrei und hängt völlig von den benutzerindividuellen Wünschen ab. Logikteile werden ausgehend von unterschiedlichen Teilen des Systems aufgerufen und verarbeitet.

3.8 PFLEGE UND ÄNDERUNG DER BENUTZERFUNKTION

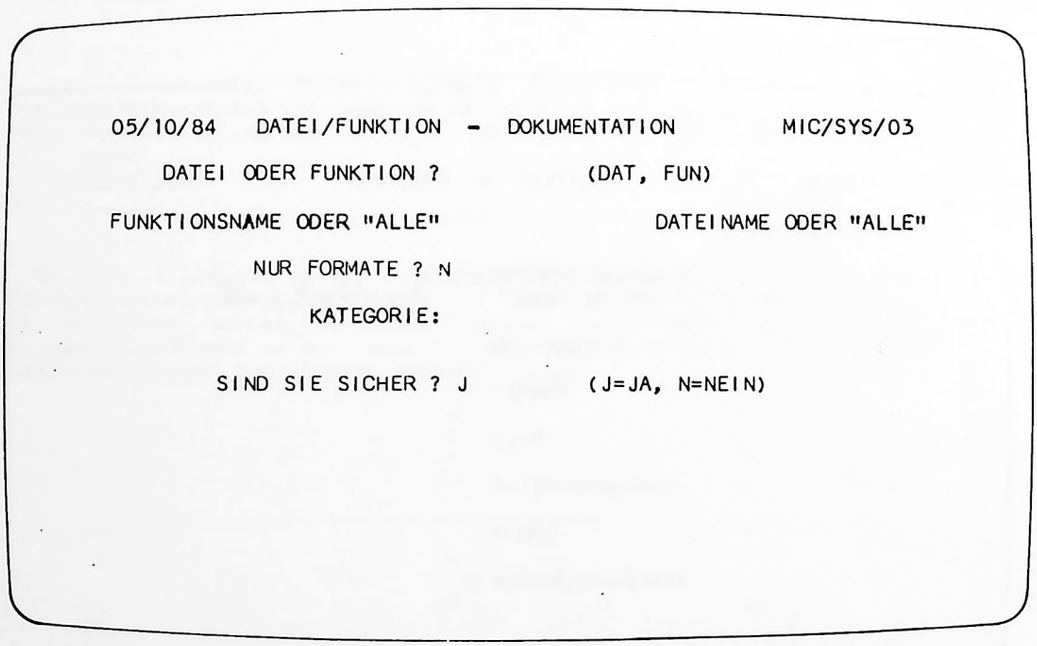
Die DEFINITIONSPFLEGE wird mit der Auswahl Nr. 4 des A*L*L Generatormenüs angesprochen und gibt dem Entwickler die Möglichkeit, ein beliebiges Definitions-Bildschirmformular anzufordern. Der Entwickler kann jeden dort gespeicherten Parameter ändern. Nach der Eingabe des Namens der Funktion, die geändert werden muß, prüft das System den Funktionsnamen. Das System stellt fest, ob auf ein Definitions-Bildschirmformular für ein Menü, für eine Bildschirmmaske, für einen Bericht oder für die Dateipflege zugegriffen werden muß. Der Benutzer erhält das gewünschte Definitions-Bildschirmformular, indem er den entsprechenden Prozeß kennzeichnet.

Nachdem die Modifikationen ausgeführt sind, braucht der Entwickler lediglich ein Kennzeichen neu zu setzen, womit die Funktion neu generiert wird.



3.9 DOKUMENTATION MIT A*L*L

Mit der Auswahl Nr. 5 des A*L*L Generatormenüs wird die Systemdokumentation angesprochen. Dieser Prozessor wird dazu verwendet, die A*L*L-Systemdokumentation für Dateien und Funktionen auszuwählen und zu drucken. Es stehen Bildschirmformulare für die Druckausgabe-Generierung von Anwendungsdefinitionen und Bildschirmmasken zur Verfügung. Die zu den Bildschirmfunktionen gehörenden Hilfe-Nachrichten werden in Verbindung mit diesen gedruckt. Die Dokumentation dient zwei wesentlichen Zwecken: einerseits für das Erstellen der Benutzerdokumentation und andererseits für das Zusammenstellen einer gedruckten Dokumentation des Anwendungssystems.



3.10 DIENSTROUTINEN DES A*L*L-SYSTEMS

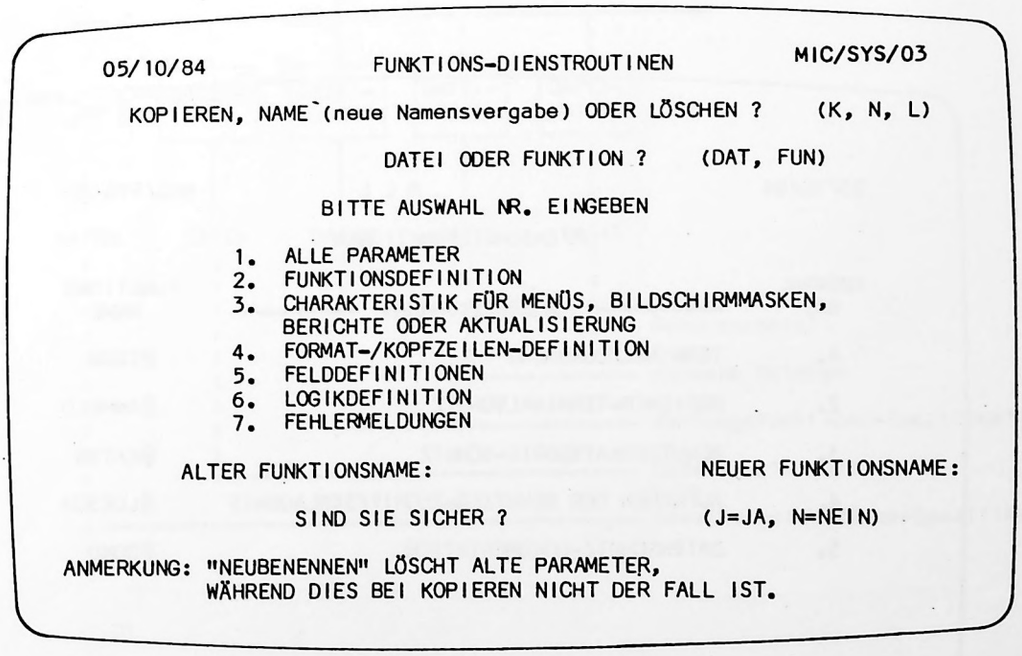
Dieser Modul wird durch die Auswahl 6 des A*L*L Generatormenüs angesprochen. Mit den Dienstroutinen ist es möglich, Datei- oder Funktionsdefinitionen innerhalb des A*L*L-Systems zu kopieren, diesen neue Namen zuzuordnen oder zu löschen. Darüber hinaus können die Dienstroutinen weiter unterteilt werden, da der Benutzer entscheiden kann, ob sich die jeweilige Dienstroutine auf nur einen Teil der Funktion oder die gesamte Funktion auswirkt.

Die KOPIER-Dienstroutine ist dafür verantwortlich, eine Datei oder Funktion unter einem vom Benutzer vergebenen Namen ausgehend von einer bereits bestehenden Datei oder Funktion zu kopieren.

Beim Einsatz der RENAME-Routine wird der Name einer bereits vorhandenen Datei oder Funktion mit einem vom Benutzer angegebenen Namen ausgetauscht.

Mit der LÖSCHEN-Dienstroutine wird eine im A*L*L-System bereits vorhandene Datei oder Funktion gelöscht.

Der Nutzen für den Anwendungentwickler ist beim Einsatz dieses Moduls in zweifacher Hinsicht gegeben. Einerseits kann der Benutzer die System-Bibliothek ohne großen Aufwand pflegen, und andererseits kann die normalerweise für die Ausführung der Entwicklungsarbeiten benötigte Zeit erheblich reduziert werden, indem wesentliche Teile der für eine neue Funktion benötigten Parameter aus einer bereits vorhandenen Funktion übernommen werden. Das bedeutet, daß bei ähnlich lautenden Funktionen zuerst eine Stammfunktion aufgebaut wird, von der aus alle anderen, individuell gestalteten Funktionen entwickelt werden.



3.11 DATENSCHUTZ-MENÜ

Das Datenschutz-Menü ist ein untergeordnetes Menü, welches durch die Auswahl 7 des A*L*L Generatormenüs angewählt wird. A*L*L arbeitet mit mehreren Datenschutz-Ebenen.

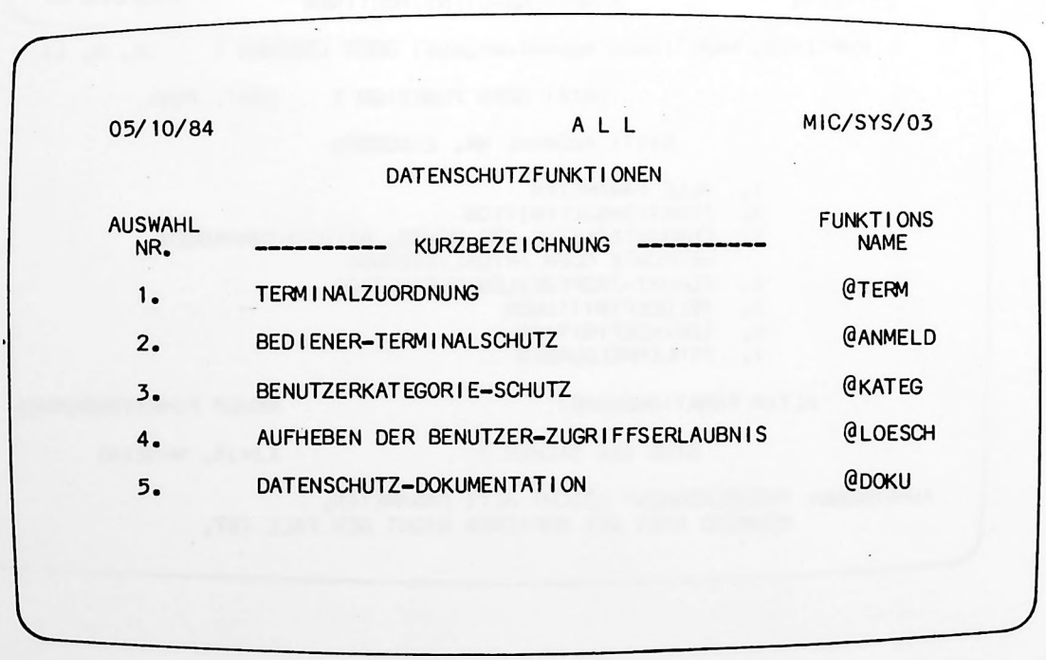
Die "TERMINALZUORDNUNG" ordnet jedem Bildschirm bzw. Terminal eine Unternehmens-/Abteilungs-Kennummer bei gleichzeitiger Angabe des Terminal-Standorts zu.

"BEDIENER-TERMINALSCHUTZ" erlaubt die Eingabe oder das Ändern des einem Benutzer zugeordneten Kennwort unterhalb der Unternehmens-/Abteilungs-Kennzahl und ordnet Termimal(s) einem Benutzer zu.

Der "BENUTZERKATEGORIE-SCHUTZ" autorisiert den Zugriff auf eine vorgegebene Datenschutz-Ebene innerhalb einer Kategorie oder Gruppe von Funktion(en) durch einen bestimmten Benutzer. Die Kategorie kann als Untersystem oder Modul betrachtet werden (z.B. Lohn-/Gehaltssystem oder Hauptbuchhaltung). Der Systemorganisator kann bestimmten Benutzern den Zugriff auf einzelne Funktionen entsprechend den individuellen organisatorischen Voraussetzungen erlauben oder eine Ebene innerhalb einer Kategorie angeben.

Mit "AUFHEBEN DER BENUTZER-ZUGRIFFSERLAUBNIS" ist es dem Systemorganisator möglich, einzelnen Benutzern den Zugriff über bestimmte Terminals und auf bestimmte Kategorien zu verwehren. Darüber hinaus ist es möglich, einem im System eingetragenen Benutzer den Zugriff auf das A*L*L-System zu verwehren, indem die Benutzerkennung vollständig gelöscht wird.

Mit "DATENSCHUTZ-DOKUMENTATION" werden alle im System gespeicherten Datenschutz-Informationen in gedruckter Form zur Verfügung gestellt.



05/10/84 A L L MIC/SYS/03

DATENSCHUTZFUNKTIONEN

AUSWAHL NR.	KURZBEZEICHNUNG	FUNKTIONS NAME
1.	TERMINALZUORDNUNG	@TERM
2.	BEDIENER-TERMINALSCHUTZ	@ANMELD
3.	BENUTZERKATEGORIE-SCHUTZ	@KATEG
4.	AUFHEBEN DER BENUTZER-ZUGRIFFSERLAUBNIS	@LOESCH
5.	DATENSCHUTZ-DOKUMENTATION	@DOKU

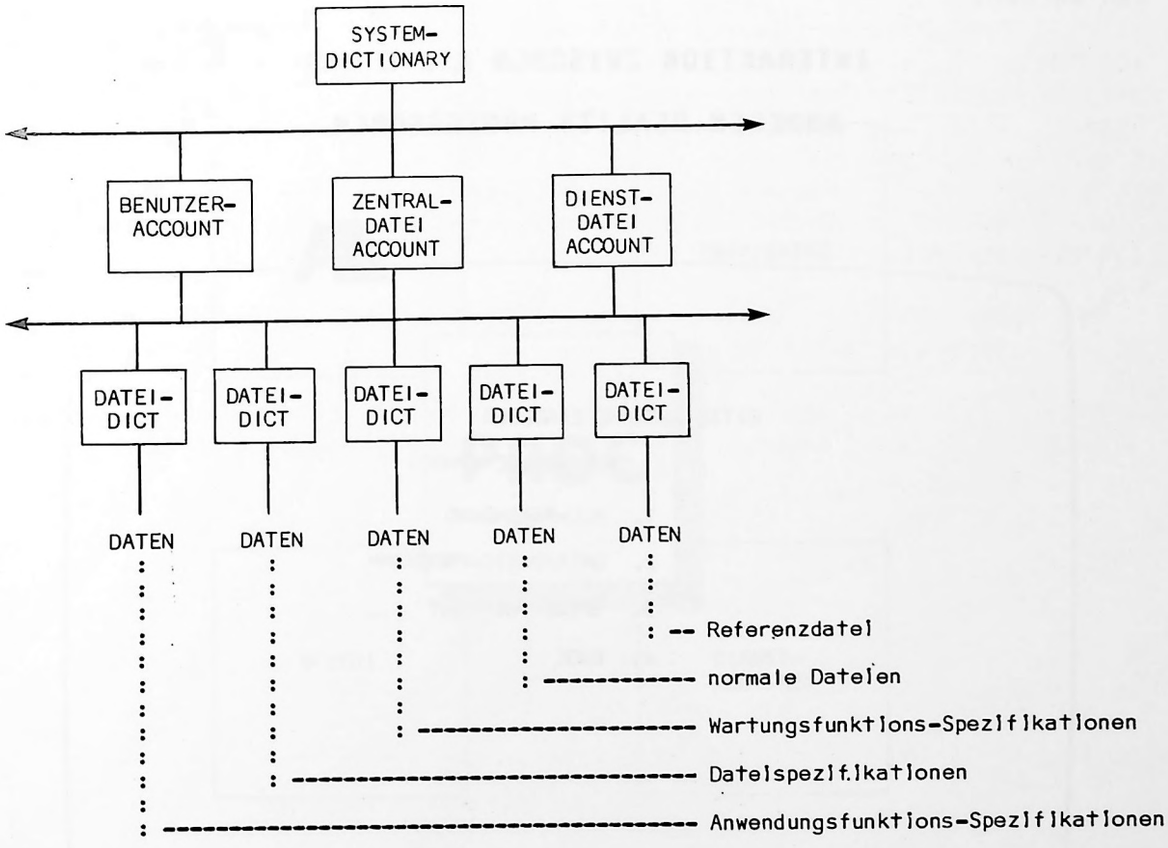
4.0 SCHNITTSTELLEN ZWISCHEN A*L*L UND ANDEREN MICRODATA-SYSTEMEN

Dieses Kapitel ist für die bereits mit den Systemen der Serien M8000 und M9000 arbeitenden Benutzer bestimmt, die nun die Leistungsfähigkeit ihres Systems mit A*L*L zu steigern wünschen.

A*L*L ist ein vollständig mit dem Betriebssystem REALITY integriertes System. Die Routinen des Systems A*L*L Ausführungssystems verwenden die in REALITY bereits vorhandenen E/A-Befehle, die Echtzeituhr-Routinen (real time clock routines) und die einzelnen Moduln des Betriebssystems. Das A*L*L-System wird innerhalb der Account-Struktur des Betriebssystems REALITY aufgesetzt, wobei die Account-Struktur durch Befehle des Betriebssystems geändert werden kann.

RELATIONALE DATENBANK

MIT EINSATZ VON A*L*L-ANWENDUNGSSYSTEMEN



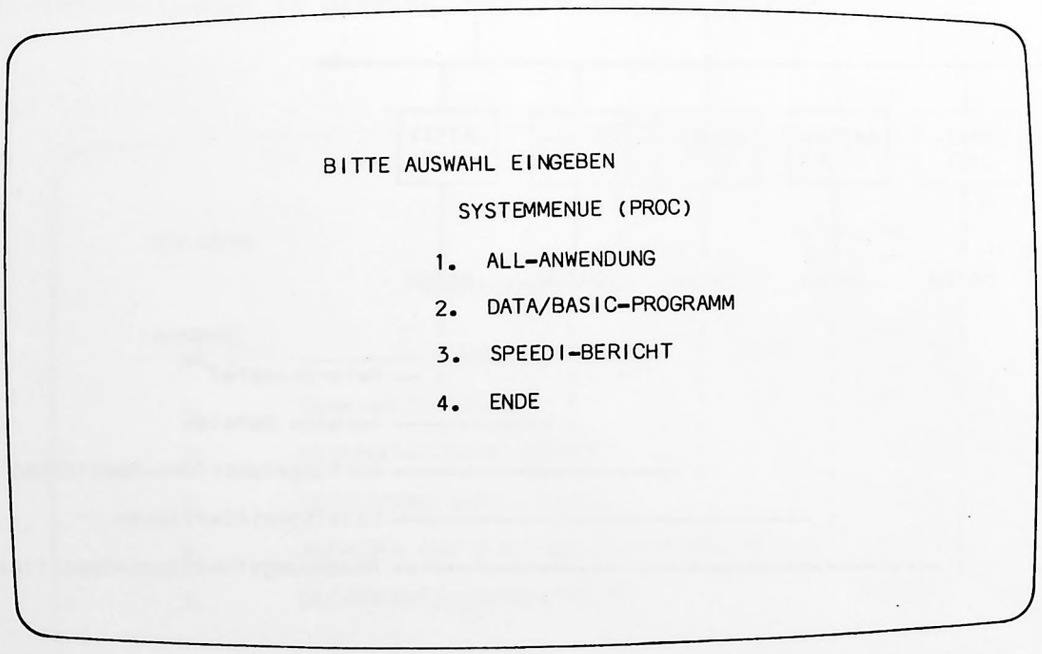
Nachdem die Dateidefinition mit A*L*L abgeschlossen ist, spricht das System den Dateierstellungs-Prozessor des Betriebssystem REALITY an. Dadurch ist es dem Benutzer möglich, für die Dateierstellung physische Plattenspeicherbereiche zu zuordnen.

Mit A*L*L ist der Entwickler in der Lage, die vom System SPEEDI verwendeten Dictionary-Einträge automatisch in die Datenbank zu speichern. Daher kann SPEEDI auch auf diejenigen Dateien zugreifen, die unter Verwendung von A*L*L erstellt wurden.

A*L*L arbeitet Hand in Hand mit den Datenschutz-Einrichtungen des Betriebssystems REALITY. Die im Betriebssystem eingebettete Account-Struktur wird beibehalten. Mit dem Verb CALL der "Terminal Control Language" (TCL) eine A*L*L-Funktion ausgeführt. CA*L*L wird in Verbindung mit den folgenden Parametern benutzt: Benutzerbezeichnung, Kennwort und Funktionname. Dadurch ist es möglich, daß eine Funktion ausgehend von TCL aufgerufen wird und A*L*L die Funktion ausführt. A*L*L-Funktionen können wie jedes andere TCL-Verb über PROC aufgerufen werden.

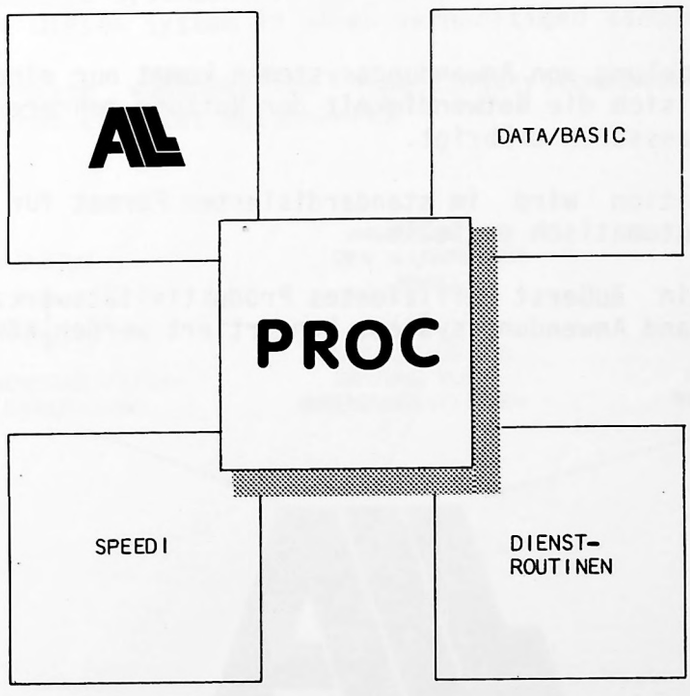
Wenn ein Benutzer bereits über DATA/BASIC-Programme, SPEEDI-Berichte und PROCS verfügt, kann A*L*L ausgehend von anderen REALITY-Prozessoren aufgerufen werden.

INTERAKTION ZWISCHEN A*L*L UND ANDEREN REALITY-PROZESSOREN



Das im Betriebssystem REALITY eingebettete SPOOLER-System kann über A*L*L aktiviert werden. Dadurch ist es dem Benutzer möglich, auf alle Einrichtungen des derzeitig installierten System-Spoolers zu zugreifen.

Es stehen PROC-Dienstroutinen zur Verfügung, die bei der Verteilung der A*L*L-Daten innerhalb des Systems oder bei der Übertragung dieser Daten zu anderen Systemen behilflich sind. Dabei kann der Benutzer bestimmte Funktions- oder Dateikategorien von einem Account zum anderen kopieren oder vom Magnetband übernehmen oder auf dieses übertragen. Diese Dienstroutinen werden ausgehend von der System-Eingabeaufforderung (:) der "Terminal Control Language" (TCL) aufgerufen. Die Dienstroutinen werden weiter wie folgt aufgeteilt: nur für das Betriebssystem zuständige Dienstroutinen und Dienstroutinen für Betriebssystem/Entwicklungssystem.



5.0 WEM NUTZT A*L*L ?

Vorteil und Nutzen dieses Systems liegen klar auf der Hand, sollten jedoch an dieser Stelle der Übersicht wegen nochmals zusammengefaßt werden. Sie sind nicht nur für Software-Hersteller sondern auch für Systemhäuser, Datenverarbeitungs-Mitarbeiter und Fachbereich-Benutzer von Bedeutung.

5.1.1 Anbieter und Systemhäuser

- Die Entwicklungszeit wird erheblich reduziert, was bedeutet, daß die Entwicklung kundenindividueller Anwendungssysteme möglich und erschwinglich ist.

- Die Vorgehensweise bei der Modifikation und Wartung der Anwendungssysteme ist äußert flexibel, was bedeutet, daß die für die Kunden entwickelten Anwendungssysteme schnell und mühelos den sich ständig ändernden Gegebenheiten angepaßt werden können.

- Das System unterstützt die Standardisierung beim Entwickeln, Implementieren, Testen, Modifizieren und Dokumentieren von Anwendungssystemen.

- Die Wartungskosten werden durch den Einsatz vereinfachter Modifizierungseinrichtungen und durch die Kontinuität der Struktur innerhalb der Systeme und über die Grenzen des jeweils entwickelten Anwendungssystems hinaus reduziert.

- Bei der Entwicklung von Anwendungssystemen kommt nur eine Methode zum Einsatz, wodurch sich die Notwendigkeit der Nutzung mehrerer Programmiersprachen oder Prozessoren erübrigt.

- Die Dokumentation wird im standardisierten Format für Anwendungssysteme und Dateien automatisch erstellt.

- A*L*L ist ein äußerst effizientes Produktivitätswerkzeug, mit dem ohne größeren Aufwand Anwendungssysteme konvertiert werden können.

5.1.2 Datenverarbeitungs-Mitarbeiter

- Durch den Wegfall von Routineaufgaben (Kodierung und Kompilierung) werden erhebliche Zeiteinsparungen erzielt, so daß es dem Entwickler möglich ist, sich wesentlich konzentrierter den Aufgaben des Gesamtentwurfs und der Entwicklung neuer Anwendungssysteme zu widmen.

- Mit A*L*L können alle anstehenden Entwicklungsaufgaben schnell und einfach erledigt werden (wie zum Beispiel der Aufbau standardisierter Bildschirmmenüs, Bildschirmmasken, Berichte und Aktualisierungs-Operationen) ohne sich dabei unterschiedlicher Prozessoren bedienen zu müssen.

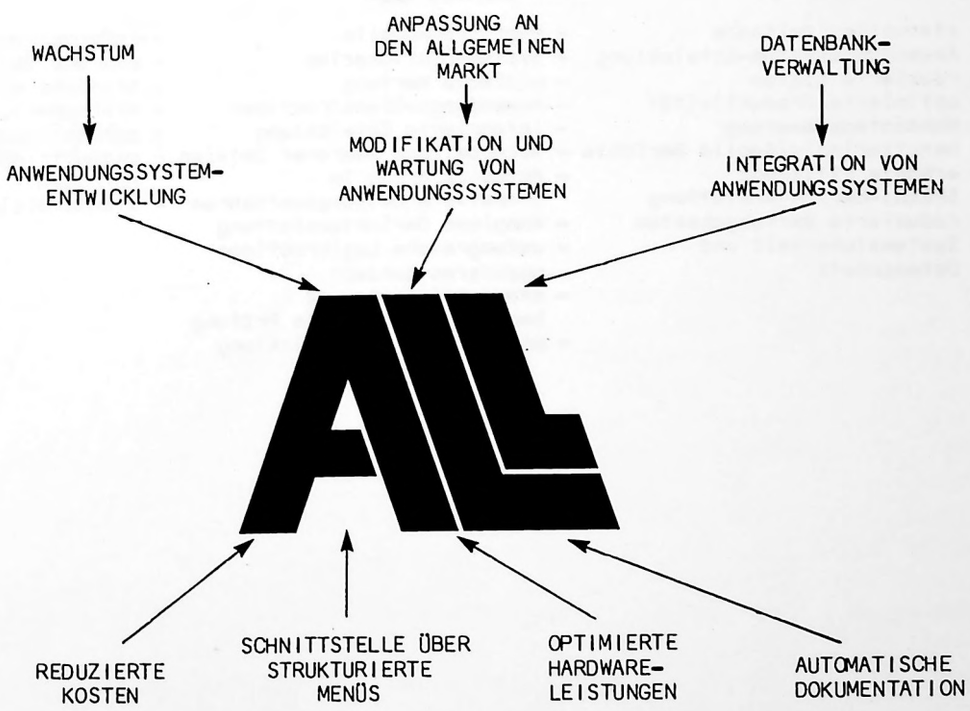
- Die Laufzeiten der mit A*L*L erstellten Anwendungssysteme sind kürzer als die manuell kodierter Programme.

- Die Systemstruktur ist gleichbleibend für alle entwickelten Anwendungssysteme, wodurch es jedem Mitarbeiter leicht fällt, die von einem anderen Mitarbeiter entwickelten Systeme zu warten und die Zielsetzungen des Systems mühelos zu verstehen.

- Die Dokumentation wird als Nebenprodukt der Anwendungssystem-Entwicklung automatisch erstellt. Die Dokumentation ist dabei immer aktuell und steht jederzeit für den Zugriff bereit.

- Die Antwortzeiten zu Systementwicklungs-Wünschen seitens der Geschäftsleitung fallen mit diesem System in einen vernünftigen Rahmen.

- Der in EDV-Kreisen wohlbekannte Anwendungssystem-Entwicklungsstau wird durch erhöhte Produktivität entschärft.



5.1.3 Fachbereichs-Benutzer

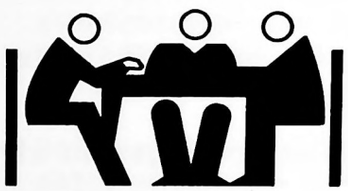
- Durch die Standardisierung der Benutzerinformationen über alle, mit A*L*L entwickelten Anwendungssysteme werden die Datenerfassungs-Aufgaben wesentlich vereinfacht.

- Die Datenprüf-, Hilfe- und Fehlermeldungen sind genau und selbsterklärend und stehen dem Benutzer während des gesamten Ablaufs des Anwendungssystems zur Verfügung.

- Anwendungssysteme werden sehr schnell entwickelt und implementiert, womit der individuelle Bedarf der einzelnen Fachabteilungen schnell abgedeckt werden kann.

- Die unter Umständen sehr schnell zu vollziehenden Systemänderungen können mühelos und zeitnah implementiert werden.

A*L*L VORTEILE



LEITENDE MITARBEITER

- sinnvolle, zeitnahe Anwendungssystem-Entwicklung
- rduzierte Kosten
- optimierte Produktivität
- Mandantensteuerung
- benutzerindividuelle Berichte
- erhöhte Effizienz
- SPEED!-Berichterstattung
- reduzierte Wartungskosten
- Systemsicherheit und Datenschutz

ANWENDUNGSSYSTEM-ENTWICKLER

- Prototyp-Modelle
- System-Dictionaries
- einfache Wartung
- Anwendungs-Dienstroutinen
- integrierte Entwicklung
- Verarbeitung mehrerer Dateien
- Aktualisierung im Stapelverarbeitungsverfahren
- komplexe Berichterstattung
- umfangreiche Logikroutinen
- modulares Konzept
- standardisierte und benutzerindividuelle Prüfung
- menügetriebene Entwicklung

BENUTZER

- interaktive Verarbeitung
- mehrere Terminals
- standardisierte Operationen
- Hilfe-Meldungen
- automatische Wiederholung
- menügetriebene Eingabe und Abfragen
- standardisierte Formate

6.0 PRODUKTPOLITIK UND FIRMENGRUNDSÄTZE

Microdata bietet exzellente Hardware und qualitativ ebenso hochwertige Software. Dadurch wird ein Vertrauensverhältnis zwischen Kunde und Hersteller geschaffen, welches dadurch erreicht wird, daß nur ein Unternehmen für beide Zweige kommerziell orientierter Computer-Produkte in der Pflicht steht. Aus diesem Grund unterhält Microdata weltweit betriebene Dienstleistungs- und Kundenunterstützung-Zentren. Mit diesen Organisationen ist Microdata für die Entwicklung und Wartung beider Produktkategorien verantwortlich - Hardware und Software.

Das Unternehmen stellt sehr hohe Beträge für Forschungs- und Entwicklungsprojekte bereit, eine Gewährleistung dafür, daß nicht nur A*L*L sondern auch alle anderen Microdata-Produkte weiterentwickelt werden.

Das Kundenschulungs-Zentrum bietet eine umfassende Schulung, die sich mit allen Aspekten der A*L*L-Systementwicklung befaßt.

Hat sich ein Kunde für den Einsatz des A*L*L-Systems entschieden, dann werden Hardware und Software zusammen mit den notwendigen Handbüchern ausgeliefert. Es stehen derzeitig die folgenden A*L*L-Handbücher zur Verfügung.

A*L*L Vorführungshandbuch

A*L*L Bedienerhandbuch

A*L*L Benutzer-Referenzhandbuch

Alle Mitarbeiter der örtlichen Microdata-Niederlassung stehen für Auskünfte und für die Kundenunterstützung bereit.

6.1 MICRODATA-SOFTWARE

6.1.1 A*L*L-Software

A*L*L kann auf den Systemen der Serien M1000, M6000, M8000 und M9000 und unter dem Betriebssystem REALITY eingesetzt werden. A*L*L wird in zwei unterschiedlichen Versionen ausgeliefert:

1. A*L*L Anwendungsgenerator (enthält Entwicklungs- und Ausführungssystem)
2. A*L*L Ausführungssystem

Der A*L*L Anwendungsgenerator ermöglicht die Entwicklung und Ausführung von Anwendungssystemen, während das A*L*L Ausführungssystem lediglich die Ausführung bereits entwickelter und generierter Anwendungssysteme gewährleistet.

6.1.2 Microdata System-Software

REALITY ist auf dem Markt eines der führenden Betriebssysteme. Das System verwaltet nicht nur alle Daten des Unternehmens in einer Datenbank, sondern es läßt auch zu, daß der Benutzer die Verknüpfungen bzw. Relationen der Datenelemente untereinander entsprechend den eigenen Bedürfnissen definiert. Die auf Dictionaries aufbauende Datenbank-Verwaltung ermöglicht es dem System, die unterschiedlichsten Bedürfnisse der Benutzer abzudecken und die Pflege der in der Datenbank gespeicherten Daten so einfach wie möglich zu gestalten.

Das Abfragesystem SPEEDI wurde speziell im Hinblick auf die Leistungsfähigkeit der von REALITY verwalteten Datenbank entwickelt. Somit ist es auch den Benutzern möglich, die keinerlei Vorkenntnisse bei der Datenverarbeitung mit Computer-Anlagen haben, schnell und einfach Berichte nach eigenen Vorstellungen zu definieren.

PROC kann dazu verwendet werden, komplexe Datenverarbeitungs-Vorgänge und Berichte bereits im Vorfeld zu definieren und zu speichern. Nach der Speicherung lassen sich diese Funktionen und Berichte unter Eingabe nur eines Befehls aktivieren.

DATA/BASIC ist eine erweiterte BASIC-Version, in der Einrichtungen enthalten sind, die die Programmierung vereinfachen und die vollständige Leistungsfähigkeit der unter REALITY verwalteten Datenbank zur Verfügung stellen.

6.1.3 Optional erhältliche Software für die Serie M9000

Es stehen die folgenden, unter dem Betriebssystem REALITY laufenden Software-Systeme zur Verfügung:

WORDMATE ist ein vollständiges Textverarbeitungs-System, welches in der Lage ist, auf die in der REALITY Datenbank gespeicherten Daten zuzugreifen. WORDMATE vereinfacht die Integration von Text und Daten erheblich.

REALCALC ist ein äußerst leistungsfähiges Spreadsheet-System (Tabellenbe- und -verarbeitung), mit welchem die in der Datenbank gespeicherten Daten gelesen und verarbeitet werden können. Damit erhalten die leitenden Mitarbeiter eines Unternehmens ein höchstfortschrittliches Werkzeug, mit dem Daten unter Verwendung von "Was ist wenn..."-Fragen analysiert und bewertet werden.

FISCAL ist als Planungs- und Analysesystem für den Aufbau komplexer Modelle konzipiert. Einem Modell können verschiedene Dateien zugeordnet werden. Mehrere Modelle können zu einem Gesamtmodell (Unternehmensplanung) zusammengefaßt werden.

Das FISCAL-Menü wird durch erläuternde Texte ergänzt, die bei Bedarf abrufbar sind. Die Modelle können jederzeit modifiziert werden.

Die Analysemöglichkeiten sind wie folgt:

- Vorwärts- und Rückwärtsrechnung,
- Sensitivitätsanalysen,
- statistische Verfahren,
- Prognoseverfahren,
- Optimierung.

Die Ausgabeformen sind wie folgt:

- Standardausgabe,
- selbstentwickelte Berichte,
- Grafik.

CPA unterstützt das Projektmanagement bei der Planung von Zeit, Kosten und Kapazitäten. CPA basiert auf den Netzplan-Verfahren CPM bzw. PERT. CPA überprüft den Netzplan auf seine formale Richtigkeit, berechnet Anfangs-, End- und Pufferzeiten (Kalenderdatum oder Zeiteinheit) und ermittelt die kritischen Vorgänge. Die Vorgangsdauer kann - zeitweise oder dauerhaft - verändert werden.

CPA erstellt den Netzplan (Grafik oder Berichtsform), das Balkendiagramm und Berichte. Zusätzliche Berichte können mit der Standard-Abfrage-Sprache erstellt werden.

Ein oder mehrere Vorgänge können in einem Teilnetzplan weiter gegliedert werden. Sämtliche Berechnungen und Berichte sind auch für den Teilnetzplan verfügbar.

Die Daten sämtlicher Vrgänge werden in einer Bibliothek gespeichert und können für die Angebotserstellung, zur Kalkulation oder für einen neuen Netzplan verwendet werden. CPA ist menügesteuert. Es besteht eine Schnittstelle zum Planungs- und Analysesystem FISCAL.

SUPERPLOT ist ein für den Benutzer erstelltes System, mit welchem Kurven-, Kreis-, Balkendiagramme, usw. erstellt werden können. SUPERPLOT wird in Verbindung mit FISCAL, CPA, der Abfragesprache SPEEDI und in BASIC-Programmen eingesetzt.

A*L*L RESULTS ist ein mit A*L*L entwickeltes, kommerziell orientiertes Buchhaltungssystem, mit welchem die Hauptbuchhaltung, Kreditorenbuchhaltung und Debitorenbuchhaltung abgedeckt werden kann.

Darüber hinaus steht eine breite Palette branchenorientierter, auf REALITY basierende Anwendungssysteme zur Verfügung, die entweder über Microdata oder unabhängige Software-Hersteller erhältlich sind.

6.1.4 System-Software der Serie M8000

Für den Einsatz im Zusammenhang mit der Serie M8000 steht eine Vielfalt unterschiedlicher Software-Systeme zur Verfügung. Dazu gehört ein fortschrittliches, mit Hochgeschwindigkeit arbeitendes Datenerfassungs-System, mit dem Datenerfassungs-Geschwindigkeiten von über 20.000 Anschlägen je Stunde über bis zu 64 Bildschirm-Terminals aufrecht erhalten werden kann. Benutzerindividuelle Anwendungssysteme können unter Verwendung der benutzerfreundlichen BASIC-Programmiersprache entwickelt werden, eine Sprache, deren Komfort zusätzlich durch den Einsatz von ECL (ENGLISH COMMAND LANGUAGE) erhöht wird. Mit ECL steuert der Benutzer den Systemablauf, die Stapelverarbeitungs-Prozesse, die Dateiverwaltung, den Ablauf von Anwendungsprogrammen und die Datenkommunikation. QUICKSTART, ein Teil von ECL, erlaubt dem Benutzer das Speichern der Job-Steuerungsbefehle, wonach es dem Benutzer möglich ist, unter Verwendung nur eines Befehls eine Ansammlung von ECL-Befehlen abzurufen und ausführen zu lassen.

6.1.5 Optional erhältliche Software-Systeme für die Serie M8000

Mit der Serie M8000 werden die folgenden Software-Systeme angeboten:

COMMUNICATIONS - es steht eine Vielfalt von Kommunikationsprotokollen für Operationen im Stapelverarbeitungs-Modus und interaktiven Modus zur Verfügung. Die folgenden IBM-Stapelverarbeitungsprotokolle sind erhältlich: 3270, 3780, 3777 SNA und HASP. Ferner sind die interaktiven IBM-Protokolle 3270, 3270 Passthrough und SNA erhältlich.

OFFICE AUTOMATION - das M8000 Büroautomatisations-Produkt enthält:

Textverarbeitung
Elektronische Post
Spreadsheet (Tabellenkalkulation)
Kalenderverwaltung
Telefonverzeichnis
Tischrechner

COBOL ANSI '74 ist ein standardisierter COBOL-Compiler, der mit den COBOL-Compilern anderer Hersteller kompatibel ist.

PERSONAL COMPUTER INTERFACE - dieses Produkt erlaubt den Anschluß eines IBM PC oder eines Personal Computers bzw. einer Arbeitsstation eines Fremdherstellers an den Rechner der Serie M8000.

7.0 A*L*L-SYSTEM - ZUSAMMENFASSUNG

A*L*L spart bei der Entwicklung, Änderung, Wartung und Dokumentation kommerziell orientierter Anwendungssysteme sowohl Zeit als auch Kosten. A*L*L ist mit Sicherheit eines der besten Produktivitätswerkzeuge zum Einsatz für die Entwicklung und Ausführung von Anwendungssystemen.

A*L*L - SPART ZEIT UND GELD
BEI DER ENTWICKLUNG, ÄNDERUNG, WARTUNG UND
DOKUMENTATION VON ANWENDUNGSSYSTEMEN

A*L*L - APPLICATION LANGUAGE LIBERATOR

Unternehmens-philosophie

McDonnell Douglas Information Systems entwickelt, baut und vertreibt Computer-Systeme der neuesten Generation mit modernster Technologie.
McDonnell Douglas Information Systems Group ist heute in über 50 Ländern vertreten und beschäftigt mehr als 12.000 Mitarbeiter.

Unterstützung

Hervorragende Computer-Systeme herzustellen ist eine Sache. Aber im täglichen Einsatz dem Kunden die nötige Unterstützung zu bieten eine andere. Denn jedes Produkt ist nur so gut wie das Unternehmen, das hinter ihm steht.
McDonnell Douglas Information Systems steht für Erfahrung, Kontinuität und Innovation in Entwicklung und Technologie. Und für Schnelligkeit und Zuverlässigkeit in der Kundenbetreuung. Leistungen, die nur einer der renommiertesten Hersteller bieten kann.

Beratung

Support- und Produktunterstützung stehen natürlich in gleicher Weise für die weitgehende Software-Unterstützung und die gute fachliche Beratung des Kunden bei veränderter Aufgabenstellung, bei Änderungen der vorhandenen Konfiguration oder bei System-Erweiterungen zur Verfügung. McDonnell Douglas Information Systems arbeitet permanent an der Verbesserung der Produkte und Leistungen. Auch Support wird bei uns als Produkt angesehen und dementsprechend beachtet.

Kundendienst

Schnelligkeit ist bei Bedarfsfällen entscheidend, um Unterbrechungen des Betriebsablaufes gering und die Kosten niedrig zu halten. Bei McDonnell Douglas Information Systems beträgt die durchschnittliche Wartezeit bei Systemstörungen nur eine Stunde. Und wir arbeiten ständig daran, diese weiter zu verringern.

Der technische Kundendienst ist rund um die Uhr einsatzbereit, sieben Tage in der Woche. Für alle Fälle. Auch wenn Sie ihn nur selten benötigen werden. Denn McDonnell Douglas Information Systems-Produkte sind äußerst zuverlässig.

Schulung

Ein wesentlicher Teil der Anwender und Bediener unserer Systeme ist beruflich auf anderen Gebieten qualifiziert als in der Datenverarbeitung. So sollte es auch sein. Die Fachkenntnisse der Anwender liegen in ihren täglichen Arbeitsbereichen – normalerweise Verwaltung, Buchführung, Handel oder Bibliothekswesen. Für diese Anwender bietet McDonnell Douglas Information Systems eine leicht verständliche Schulung in einem gut ausgerüsteten Schulungszentrum an, wobei viel Wert darauf gelegt wird, daß die verwendete Sprache vom Kunden verstanden wird. Kunden, die eine größere Anzahl von Schulungsteilnehmern haben, stellt McDonnell Douglas Information Systems einen Ausbilder zur Seite, der vor Ort die entsprechenden Lehrgänge absolviert, die dem Standard im Ausbildungszentrum entsprechen.

MCDONNELL DOUGLAS INFORMATION SYSTEMS, GMBH COMPUTER SYSTEMS DIVISION

McDonnell Douglas
Information Systems GmbH
Computer Systems Division
Hahnstraße 70, Lyoner Stern
6000 Frankfurt/Main 71
Telex 416505 micro d
Postfach 710664
Telefon 069/66416-0
Telefax 069/66647 03

Geschäftsstellen:

McDonnell Douglas
Information Systems GmbH
Sportallee 6
2000 Hamburg 63
Tel.: 040/510001

McDonnell Douglas
Information Systems GmbH
Lürmannstraße 21
2800 Bremen
Tel.: 0421/349581

McDonnell Douglas
Information Systems GmbH
Konrad-Adenauer-Straße 25
5000 Köln 50
Tel.: 0221/351071

McDonnell Douglas
Information Systems GmbH
Hahnstraße 70, Lyoner Stern
6000 Frankfurt/Main 71
Tel.: 069/66416-0

McDonnell Douglas
Information Systems GmbH
Heilbronner Straße 314
7000 Stuttgart 30
Tel.: 0711/814511

McDonnell Douglas
Information Systems GmbH
Ingolstädter Straße 20-22, BN 2
8000 München 45
Tel.: 089/3596004

McDonnell Douglas
Information Systems S.A.
Ruetistraße 14
CH-8952 Schlieren
Telefon 01-7301873
Telex 827740

McDonnell Douglas
Information Systems S.A.
Ruetistraße 14
CH-8952 Schlieren
Tel.: 01-7301873

McDonnell Douglas
Information Systems S.A.
28 Boulevard du Pont D'Arve
CH-1205 Geneva
Tel.: 022-214235